Analysis report

Hybrid recommendations for automatic adaptive authoring in AR-maintenance knowledge capture applications

Abstract

This document presents the experimental results and analyses that contribute to the research validation of *"Hybrid recommendations for automatic adaptive authoring in AR-maintenance knowledge capture applications"*. The following packages, functions and themes have been used for data analysis within the R code presented in this document.

```
# Functions to manage and analyse data
library(dplyr)
library(tidyr)
library(car)
library(scales)
# Functions to work with plots
library(ggplot2)
library(grid)
library(ggpubr)
library(gridExtra)
# Functions to work with tables
library(knitr)
library(kableExtra)
# Additional functions
source('Code.R')
# Declare colour palettes
c03Palette = c("#1A406A","#7F7F7F","#0D1930")
c06Palette = c("#3F97C0","#1A406A","#9EBF43","#C2446F","#F2BC41","#5D4184")
c09Palette = c("#3F97C0","#1A406A","#9EBF43","#C2446F","#F2BC41","#5D4184",
               "#D32D40","#7F7F7F","#0D1930")
c12Palette = c("#D32D40","#F2BC41","#9EBF43","#3F97C0","#5D4184","#C2446F",
               "#791C24","#C77F3A","#617628","#2A386B","#402D55","#782A43")
# Declare plot theme
plotTheme <-
    theme(panel.background = element_rect(colour= "gray90", fill = "white"),
          strip.background = element_rect(colour = "gray90", fill = "white"),
          panel.grid.major = element_line(colour = "gray90", size = 0.35),
          panel.grid.minor = element_line(colour = "gray90", size = 0.175),
          axis.ticks = element_blank(),
          text = element_text(size = 11, family = "Times"))
```

This document is structured as follows. Its **first section** introduces this research's validation aim and objectives. The validation protocol, including research hypotheses and methods along with their criteria and protocols, is presented in the **second section**. Based on the outcomes of these experimental protocols, the **third section** describes their quantitative and qualitative analysis. Its **fourth section** describes the analysis results and discusses their impact on the research validation. Finally, the **fifth section** draws the research's conclusions, along with the analysis assumptions made to infer those.

1. Methodology

The research *"Hybrid recommendations for automatic adaptive authoring in AR-maintenance knowledge capture applications"* proposes recommendable content formats for automatic adaptive authoring in AR-maintenance knowlegde capture applications. These aim to improve efficiency of maintenance reporting applications by (1) enhancing data input through instantiable augmented content and (2) data selection through hybrid (context-aware and ontology-based) recommender facets. These have been implemented for a case of study in diagnosis reporting. Hence, this research's validation should aim to evaluate the impact of content formats and recommendation facets in efficiency of maintenance reporting operations.

Since reporting operations are mostly human tasks, evaluation of human-computer interaction technologies should focus on their impact on human performance. Common methos in academia [**ref**] to evaluate impact on human performance measure quantitative and qualitative criteria regarding their impact on efficiency (**time**), effectiveness (**errors**) and **usability**. **Workload** assessment has also been found useful [**ref**] when human tasks do not solely depend on manual performance, but also in other elements like temporal or mental demand. Qualitative **workload** evaluation helps to identify the nature of tasks being performed to contextually analyse efficiency and usability measurements [**ref**]. Besides, this research's contributions proposes recommendations that can affect huamn performance in reporting operations. So, the **accuracy** of those recommendations should also be analysed [**ref**].

This report aims to analyse and discuss the validity of the following research's contributions:

1. Proposed recommendation facets can improve identification (**recommendations accuracy** and **workload**) of fault conditions in diagnosis scenarios.
2. Proposed recommendable content formats can improve human performance (**time**, **errors** and **usability**) in diagnosis reporting operations.

The following section presents the resultant research hypothesis and methods conducted to validate the research objectives stated above.

2. Design

Inspired by similar research [**Wang, Gimeno and Hung**], this research's validation proposes experimental and survey methods to evaluate the abovementioned research contributions. Table 1 presents these methods, the criterions they aim to analyse and the objectives for doing so.

Table 1: Overview of validation methods, criterions and objectives.

Method	Quantitative	Qualitative	Objective
Experiments	Recommendations accuracy		Evaluate the proposal's ability to produce recommendations for indentifying faulty conditions in diagnosis scenarios
Experiments	Reporting time		Evaluate the proposal's ability to reduce errors for improving effectiveness of diagnosis reporting operations
Experiments	Reporting errors		Evaluate the proposal's ability to reduce time for improving efficiency of diagnosis reporting operations
Surveys		Reporting usability	Evaluate the proposal's perceived usability to enhance semantic understanding of diagnosis reporting operations
Surveys		Reporting workload	Evaluate perceived workload of diagnosis reporting operations

For these methods and criterions to appropriately evaluate diagnosis reporting effects, the following assumptions must hold true:

- Implemented recommendation facets select component conditions that can be considered faults of the contextual failure. So, accurate recommendations and their selection can affect diagnosis reporting performance through their ability to simplify reporting tasks.
- Reporting operations consist of data input steps. So, errors or incorrectly inputted values can be considered a measure of reporting effectiveness. If the above is true, then time and errors results should not be correlated.
- Reporting operations are human tasks. If errors can be considered a measure of reporting effectiveness, reporting time can be considered a measure of reporting efficiency.
- Reporting tools usability can affect human performance if it is not compatible with reporting tasks requisites like temporal, mental or physical demand.

Validation experiments require additional reporting tools to which compare this research's proposals. These tools or **solutions** should have different attributes regarding this research's contributions (content formats and recommendations). As part of this validation, the authors developed the following tools for comparison:

- RPMAU (ARR): the proposed AR solution that includes content formats and recommendations for knowledge capture applications.
- PMAU (ARN): an alternative AR solution that includes content formats for knowledge capture applications but not recommendations.
- Web-based recommendable reporting (TBR): an alternative non-AR solution that includes recommendations for knowledge capture applications.
- Web-based reporting (TBN): an alternative non-AR solution that does not include recommendations for knowledge capture applications.

The following subsections describe these methods and their hypotheses as well as the case of study, experimental scenarios and the tested sample.

2.1. Stopwatch time, errors and accuracy studies

Stopwatch **time**, **errors** and **accuracy** studies aim to analyse the effect of the proposed authoring **solution** (ARR) on reporting effectiveness and efficiency compared to alternative solutions (ARN, TBR, TBN) in different **failure** conditions (Electric and Electronic). Stopwatch studies consist of testers performing diagnosis reporting procedures regarding different **failure** in which collect quantiative data regarding the abovementioned criterions. In order to validate this research's contributions, these studies evalute the following hypotheses:

- Recommendations **accuracy** improves with the use of AR-content compared to non-AR recommendable solutions.
- Reporting **errors** reduce with the use of AR-content compared to non-AR reporting solutions.
- Reporting **errors** reduce with the use of recommendations compared to non-recommendable solutions.

- Reporting **time** decreases with the use of AR-content compared to non-AR reporting solutions.
- Reporting **time** decreases with the use of recommendations compared to non-recommedable solutions.

Table 2 defines the quantitative variables relevant in these studies.

Table 2: Description of measured response and factor variables in stopwatch studies.

Variable	Type	Definition
Time	Response	Number of seconds taken by a tester to complete a diagnosis reporting step of a given failure
Errors	Response	Number of mistakes when inputting failure data made by testers when conducting diagnosis reporting steps
Accuracy	Response	Number of times reporting fault condition is recommended and selected by a tester in a diagnosis reporting step
Report	Factor	Ontology class instantiated to report part of the failure's root cause
Failure	Factor	Unexpected or incorrect asset behaviour that triggers a diagnosis reporting step
Solution	Factor	Reporting tool utilised by a tester to conduct a diagnosis reporting step

Each factor variable has been given different levels to evaluate validity of these studies' hypothesis. Table 3 declares these factors' levels.

Table 3: Description of relevant factors' levels in stopwatch studies.

Factor	Level	Definition
Report	Step	Action aimed to identify a fault or that is required for fault identification
Report	EvaluatedState	Condition of component behaviour being hypothesised for determining a fault
Report	DiagnosisState	Condition of component behaviour to which compare hypothesis for determining a fault
Failure	CNN	Electrical failure to which the experimented asset is setup to for testers to conduct diagnosis reporting procedures
Failure	TEM	Electronic failure to which the experimented asset is setup to for testers to conduct diagnosis reporting procedures
Solution	ARR	Proposed AR solution that includes content formats and recommendations for knowledge capture applications
Solution	ARN	Alternative AR solution that includes content formats for knowledge capture applications but not recommendations
Solution	TBR	Alternative non-AR solution that includes recommendations for knowledge capture applications
Solution	TBN	Alternative non-AR solution that does not include recommendations for knowledge capture applications

Stopwatch experiments aim to test different reporting tools in various diverse diagnosis reporting procedures. Figure 1 presents an example of a tester's experimental procedure to report a failure's root cause. It consists of three reporting tasks, each of which comprises an ontology individual to instantiate (*'Step'*, *'evaluatesStep'*, and *'diagnosisStep'*).

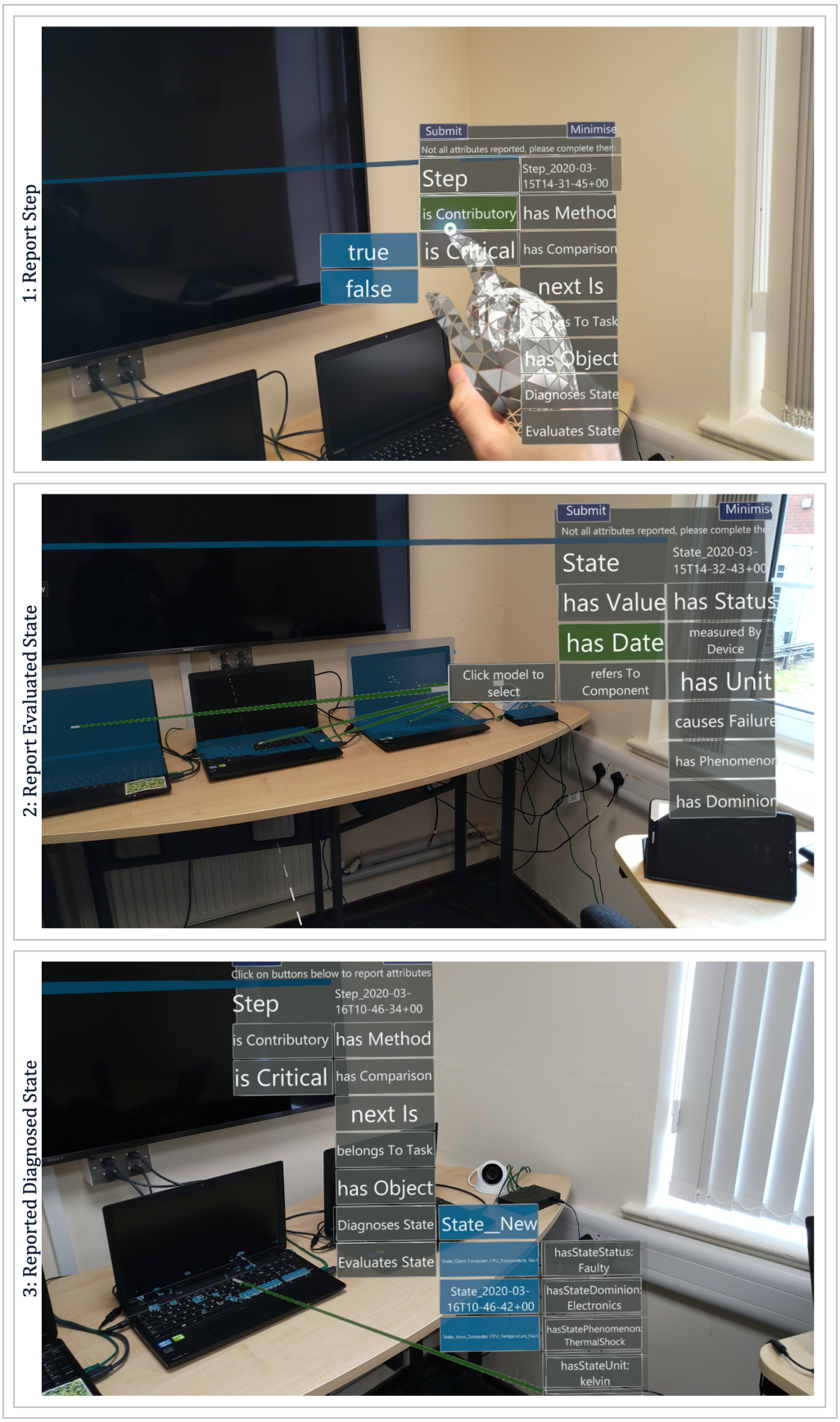


Figure 1: Examples of recommendable fabrications in fault diagnosis reporting experiments.

For coherency of further statistical analyses, testers were allocated in different groups resulted from combinations of factor levels. Table 4 presents these groups.

Table 4: Description of stopwatch experimental groups.

X	ARR	ARN	TBR	TBN
CNN	A	C	D	B
TEM	B	D	C	A

The reason to re-use testers with two different solutions was for allowing them to compare the usability between AR and non-AR solutions. This is comparison is necessary because testers are **assumed** to have none or very little previous experience in maintenance or AR. Besides, the diagnosed failures (described in **Section 2.3** can be **considered** sufficiently different in nature for not expecting carry-over effects between experiments.

2.2. Usability and workload surveys

Usability surveys aim to evaluate the perceived validity of the proposed AR methods to report diagnosis information compared to alternative solutions. Usability refers to the ability of a reporting tool to submit information regarding the diagnosis operation being conducted. Usability is a feature perceived by users and so, subject to opinion. Based on similar research [**refs**], the criteria used to evaluate usability in these surveys is that presented by Nielsen in his 1993 book "Usability Engineering" [**ref**]. These usability criterions aim to evaluate different aspects of the proposed solution regarding its formats and its impact on diagnosis reporting operations. Table 5 defines these criterions and the solution's aspects they refer to.

Table 5: Description of criterions and aspects in usability surveys.

Criterion	Aspect	Scale
Ease-to-learn	Start, Finish, Intuitiveness	Likert 1-5
Ease-to-use	Buttons-Gestures, Keyboard-Dictation, Text	Likert 1-5
Accuracy	Overlay, Shaking, Occlusion, Visualisation, Latency	Likert 1-5
Effectiveness	Efficiency, Confidence	Likert 1-5
Satisfaction	Design, Feeling, Overall	Likert 1-5

Each criterion includes a separate survey section with several statements for each aspect regarding the reporting solutions tested in stopwatch experiments. Testers were asked to determine their agreement with these statements in a Likert Scale (1-5). The results collected serve to evaluate the proposed AR methods' usability compared to other specific approaches.

Workload surveys aim to evaluate testers perceived performance requisites regarding reporting experiments for contextually analyse time and errors experimental results. In order to evaluate perceived reporting performance, the authors employed the NASA Task Load indeX (NASA-TLX) surveys. NASA-TLX is a standard questionnaire developed by NASA Ames Research [**ref**] for collecting workload self-evaluation results from experimental testers. It is a testers' self-rating procedure that provides an overall workload score based on six weighted aspects. Table 6 defines these workload factors.

Table 6: Description of workload factors employed in NASA-TLX surveys.

Workload.aspect	Definition
Mental Demand	How much mental and perceptual activity is required (e.g. thinking. deciding, calculating, remembering. looking, searching, etc)? Is the task easy or demanding, simple or complex, exacting or forgiving?
Physical Demand	How much physical activity is required (e.g. pushing, pulling, turning. controlling, activating, etc.)? Is the task easy or demanding, slow or brisk, slack or strenuous restful or laborious?
Temporal Demand	How much time pressure do you feel due to the rate or pace at which the tasks or task elements occurr? Is the pace slow and leisurely or rapid and frantic?
Performance	How successful do you think you are in accomplishing the goals of the task set by the experimenter? How satisfied are you with your performance in accomplishing these goals?
Effort	How hard do you have to work (mentally and physically) to accomplish your level of performance?
Frustration	How insecure, discouraged, irritated. stressed and annoyed versus secure, gratified, content, relaxed and complacent do you feel during the task?

Workload surveys consist of two, self-rating steps. Prior to experiments, testers are asked to evaluate the relative importance of each workload aspect given an understanding of the tasks to perform. After the experiments, testers are asked to complete a second questionnaire to quantitatively evaluate the importance of each aspect independently. These help to provide weighted scores for each workload aspect for contextually analysing experimental results regarding performance effectiveness and efficiency.

Protocols to collect and analyse experimental and survey data are described in **Section 2.5**. Instead, the following section presents the experimental cases of study along with the testing population's sample.

2.3. Case of study

The case of study comprises two diagnosis reporting procedures of no-fault-found scenarios in a complex-engineering asset. This case of study is based on expert interviews already discussed at [**ref**]. Figure 2 presents a picture of the case study's asset. The asset, named Helicopter Mission System (HMS), is a replica of an electronic system whose aim is to control the navigation mission of a helicopter. This replica was built with the same specifications as the original in order to enable laboratory experimentation. This system comprises three computers, one camera and an ethernet switch to connect them altogether. The first computer, called 'main mission computer', is used as controller for the rest of the elements and also controls the navigational parameters of the helicopter. The second computer, or 'client mission computer', is that used by helicopter pilots set the navigation mission. The third computer, which acts merely as a 'monitor', is that from which pilots control the 'client mission computer'. The 'camera' is there to provide pilots with a visual of the terrain while handling the helicopter. Finally, the 'ethernet switch' aims to connect the main mission computer to the client mission computer, the 'monitor' and the 'camera' for further control. The system comes with an integrated control monitoring system that evaluates electronic performance parameters from its different elements. Due to its criticality for piloting the helicopter, real-life maintainers are very careful when reporting diagnosis procedures on it. Besides, the control monitoring has some limitations regarding the electronic parameters it can control due to the system's configuration.

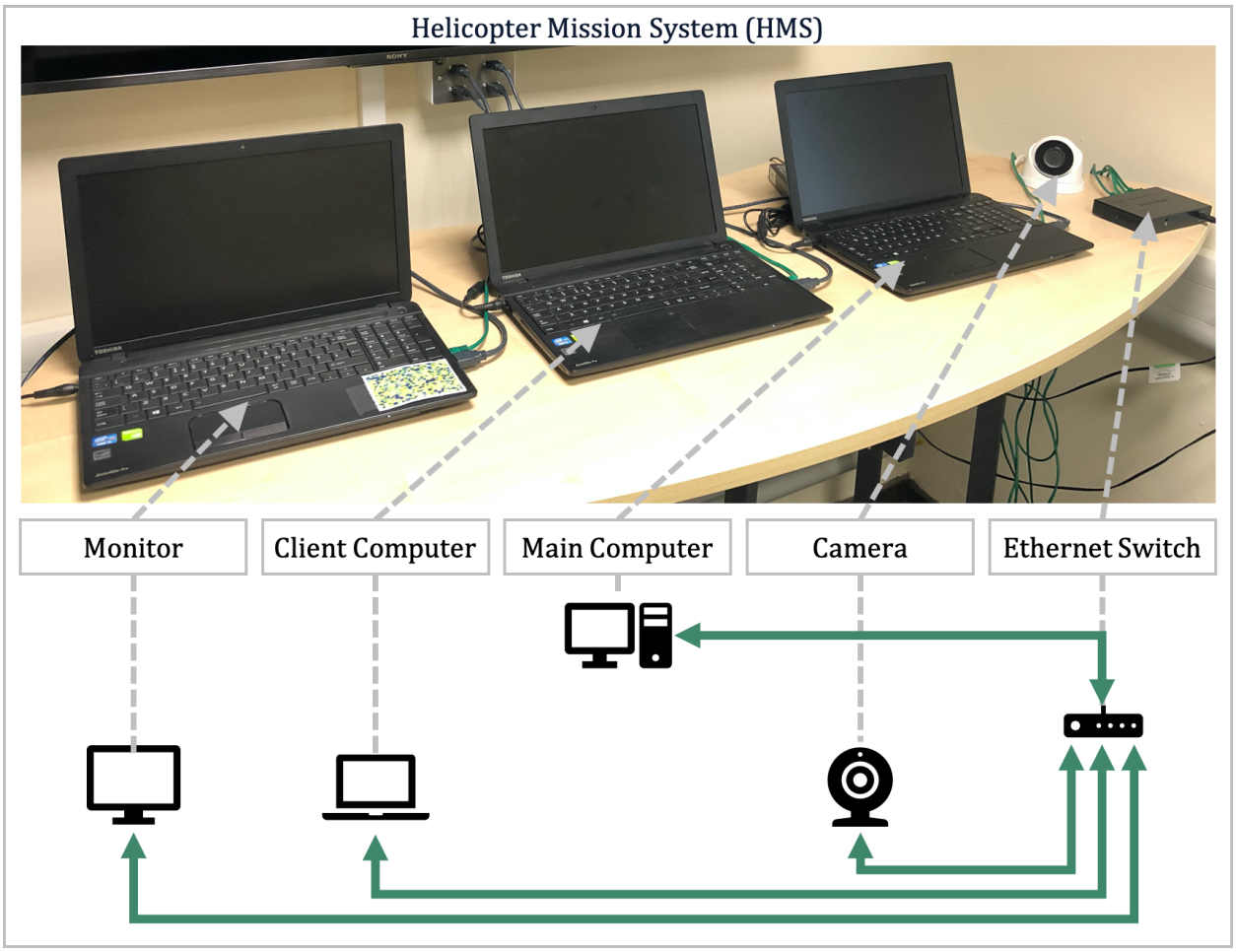


Figure 2: Overview of diagnosis reporting case of study: Helicopter Mission System (HMS).

These system's complexities make them suitable for this research's experimental methods. The system is prone to incur in no-fault-found scenarios more often than other systems in the helicopter asset, which maintainers are asked to report in details. Interviews with expert maintainers [**ref**] allowed to identify two common no-fault-found scenarios that could be uses as experimental procedures. The following subsections explains these failures and their reporting procedures.

2.3.1. Electric failure reporting scenario (CNN)

Figure 3 describes the most mentioned electronic failure in expert interviews regarding the HMS. The failure is caused by a fault in the cable that connects the "main computer" with the "ethernet switch". This failure provokes a no-fault-found condition because that cable cannot be monitored by the system's control module. The control module is managed by the "main computer" and one of its limitations is that it cannot evaluate its own connectivity to the "ethernet switch". So when this disconnection occurs, the control module shows the rest of the system's components ("client computer", "monitor" and "camera") as not connected even though the connectors are in good condition. As shown in Figure 3, connectivity is measured by connectivity time, which is identified as zero by the control module when there is no connection. The experimental reporting steps are also shown in Figure 3. Testers were asked to report the failure's root cause (blue), including a 'Step' and its 'evaluated' and 'diagnosed' 'States'.

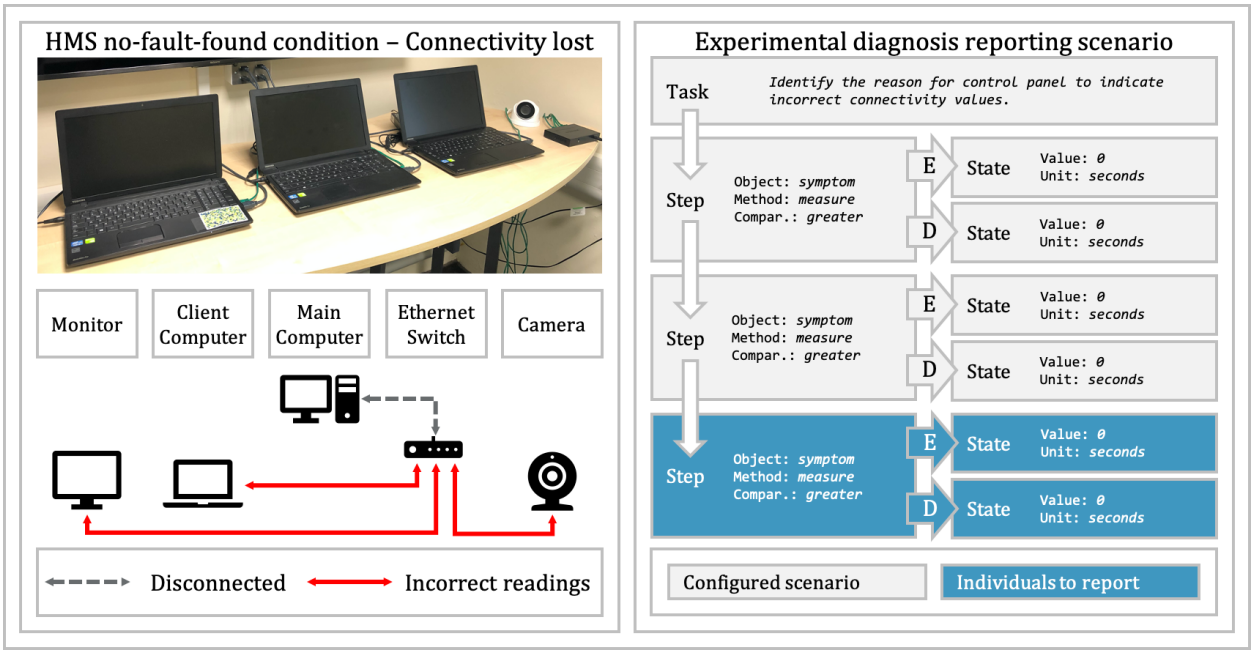


Figure 3: First case of study diagnosis reporting experiment: computers' connectivity (CNN).

2.3.2. Electronic failure reporting scenario (TEM)

Figure 4 describes the most mentioned electric failure in expert interviews regarding the HMS. It is another example of no-fault-found because the thresholds established by the system's control module are higher than the values that cause the failure. The failure consists of a hardware overload caused by too many software applications being run in the HMS simultaneously. When this occurs, both "main computer" and "client computer" reach CPU temperatures higher than 60o Celsius (~333 Kelvin). However, the system's control module cannot detect this issue because the CPU temperatures monitoring thresholds are set for each CPU independently at a temperature of 90o Celsius. Similarly to previous failures, Figure 4 shows a simplified version of the failure report including the steps to be reported by experimental testers (blue).

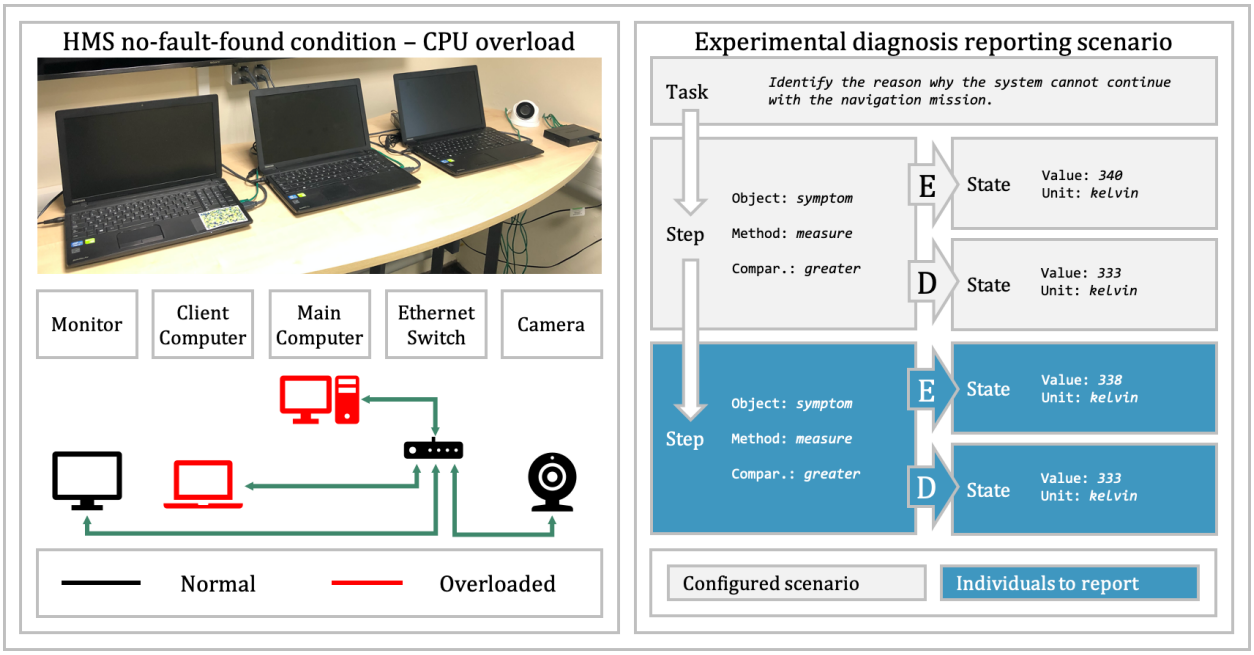


Figure 4: Second case of study diagnosis reporting experiment: CPUs temperature overload (TEM).

2.4. Experimental sample

A total of 30 MSc students (19 males and 9 females) participated as testers in laboratory experiments. Their ages range from 21 to 30 years and they are all enroled in engineering-related MSc degrees. Although they have some basic knowledge in AR and maintenance due to their courses, they have no previous hands-on experience in any of them. So, they were given a short training on AR devices right before experimentation to avoid the presence of any learning curves. Testers were randomly allocated to one of the four groups (A (7), B (7), C (7) or D (7)) to avoid "carry-over" effects between failures while using two different reporting solutions.

2.5. Experimental protocol

The protocol comprises the steps to collect and analyse experimental and survey data for validating this research proposal against its expected contributions. The abovementioned validation methods in the case study contexts described above. The following list summarises this protocol: 1. Data collection (30 testers per experiment): a. AR-maintenance introduction: to briefly train testers on the purpose of experiments, the use of reporting solutions and the experimental failures to report. b. Stopwatch time, errors and accuracy experiments: to capture quantitative data on the effect on effectiveness and efficiency of reporting in different diagnosis reporting operations. c. Usability and Workload surveys: to capture qualitative data on tester's opinions regarding usability of reporting tools and workload of reporting operations. 2. Data analysis (30 testers per analysis): a. Recommendations accuracy study: to evaluate the correctness of recommendations for describing fault conditions in reporting diagnosis and its impact in diagnosis reporting. Results should refect that there is a significant difference in recommendations accuracy between hybrid (ARR) and ontology-based (TBR) recommender methods. Graphical analysis and t-tests will be used for this matter. b. Errors effect study: to evaluate the effect on reporting effectiveness of recommender and AR methods in failure reporting procedures. Results should reflect a significant differences in reporting errors between AR (ARN, ARR) and non-AR (TBN, TBR) reporting tools. They should also reflect a significant difference between recommendable (ARR,TBR) and non-recommendable (ARN,TBN) solutions. Due to the number of experimental factors (Failure, Solution), a two-way ANOVA test will be used to test these hypotheses. Additional post hoc

comparisons (TukeyHSD test) will help to further analyse existing interactions between factors. c. Time effect study: to evaluate the effect on reporting efficiency of recommender and AR methods in failure reporting procedures. Results should reflect that AR (ARR,ARN) solutions' results are significantly different to non-AR (TBN,TBR) results. There should also be significant differences in AR and non-AR solutions between those that implement (ARR,TBR) and do not implement (ARN,TBN) recommendations. Similarly to errors study, two-way ANOVA and TukeyHSD tests will be used to validate these hypotheses. Besides, Pearson's coefficient will be used to evaluate the correlation between time and errors results for testing the assumption that errors measure effectiveness and time measures efficiency. d. Workload study: to quantitatively evaluate the relevancy of workload requisites in diagnosis reporting procedures. Results should help to contextualise previous experimental results analyses and refute the differences between time and errors. Basic statistics and graphical analyses will be used to analyse workload results. e. Usability study: to quantiatively evaluate testers' opinions on reporting tools' usability. Results should reflect improved perceived usability for those tools that implement recommendations and AR content (ARR,ARN) for indicating validity of effectiveness and efficiency improvements previously tested. Basic statistics and graphical analyses will be used for this matter.

This experimental protocol aims to validate this research's proposed methods against its expected contributions. For this validation to be coherent, there are few assumptions to consider: - In order to keep consistency within experiments, these were conducted in a laboratory environment to maintain constant other factors (e.g. ergonomics or lighting conditions) that may affect the results. Hence, these factors were considered out of these experiments' scope. - Experimental sample size for the abovementioned statistical tests can be estimated "a priori". Such estimation can be done using a F test for the most requiring analytical test (two-way ANOVA). With 4 factor groups (failure and solution), a variance of 0.3 (partial eta squared), a type-I error of 0.1 (alpha) and a power of 0.9 (1 – beta), the resultant sample size is 31 people. That is quite close to the 28-sample size used in these experiments. Besides, these numbers are similar to those achieved by similar researches [**refs**] (30-sample size). - As described above, testers are MSc students with none or very little experience in AR or maintenance. Although this ensures a baseline for measuring reporting effectiveness and efficiency, further experiments should be required to corroborate laboratory results in real-life working conditions with real maintainers to ensure validity of these hypotheses.

This protocol's results are discussed in the following section.

3. Analysis

3.1. Data collection, formatting and pre-processing

Each data set has been prepared in R-readable formats (long tables) for further treatment. These data sets can therefore be imported and transformed into data frames.

```
## 'data.frame':    28 obs. of  5 variables:
##  $ Tester  : Factor w/ 28 levels "1","4","5","6",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Failure : Factor w/ 2 levels "CNN","TEM": 2 1 1 2 2 1 1 2 2 1 ...
##  $ Solution: Ord.factor w/ 2 levels "TBR"<"ARR": 2 2 1 1 2 2 1 1 2 2 ...
##  $ Report  : Factor w/ 1 level "Step": 1 1 1 1 1 1 1 1 1 1 ...
##  $ Accuracy: int  0 1 0 1 1 1 1 0 1 1 ...
```

```
## 'data.frame':    168 obs. of  8 variables:
##  $ Tester  : Factor w/ 28 levels "1","4","5","6",..: 1 1 1 1 1 1 2 2 2 2 ...
##  $ Failure : Factor w/ 2 levels "CNN","TEM": 2 2 2 1 1 1 2 2 2 1 ...
##  $ Solution: Ord.factor w/ 4 levels "TBN"<"TBR"<"ARN"<..: 4 4 4 1 1 1 1 1 1 4 ...
##  $ Report  : Factor w/ 3 levels "DiagnosesState",..: 3 2 1 3 2 1 3 2 1 3 ...
##  $ Errors  : int  1 1 NA 2 1 1 5 2 1 0 ...
##  $ Correct : int  6 7 NA 5 7 7 2 6 7 7 ...
##  $ Total   : int  7 8 NA 7 8 8 7 8 8 7 ...
##  $ Percent : num  0.14 0.13 NA 0.29 0.13 0.13 0.71 0.25 0.13 0 ...
```

```
## 'data.frame':    168 obs. of  5 variables:
##  $ Tester  : Factor w/ 28 levels "1","4","5","6",..: 1 1 1 1 1 1 2 2 2 2 ...
##  $ Failure : Factor w/ 2 levels "CNN","TEM": 2 2 2 1 1 1 2 2 2 1 ...
##  $ Solution: Ord.factor w/ 4 levels "TBN"<"TBR"<"ARN"<..: 4 4 4 1 1 1 1 1 1 4 ...
##  $ Report  : Factor w/ 3 levels "DiagnosesState",..: 3 2 1 3 2 1 3 2 1 3 ...
##  $ Seconds : int  155 86 NA 215 163 173 188 204 195 133 ...
```

```
## 'data.frame':    168 obs. of  6 variables:
##  $ Tester      : Factor w/ 28 levels "1","4","5","6",..: 1 1 1 1 1 1 2 2 2 2 ...
##  $ Criterion   : Ord.factor w/ 6 levels "Mental Demand"<..: 1 2 3 4 5 6 1 2 3 4 ...
##  $ Weight      : int  4 0 3 4 3 1 1 0 2 3 ...
##  $ RawRate     : int  8 4 2 14 4 8 10 2 6 14 ...
##  $ AdjustedRate: int  32 0 6 56 12 8 10 0 12 42 ...
##  $ WeightedRate: num  2.13 0 0.4 3.73 0.8 ...
```

```
## 'data.frame':    784 obs. of  5 variables:
##  $ Tester   : Factor w/ 28 levels "1","4","5","6",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ Solution : Ord.factor w/ 4 levels "TBN"<"TBR"<"ARN"<..: 4 1 4 1 4 1 1 4 4 1 ...
##  $ Criterion: Ord.factor w/ 5 levels "Ease-To-Learn"<..: 1 1 1 1 1 1 2 2 2 2 ...
##  $ Aspect   : Factor w/ 14 levels "Buttons-Gestures",..: 12 12 5 5 8 8 1 1 9 9 ...
##  $ Response : int  3 4 4 3 4 4 5 4 5 4 ...
```

Modifications:

- Data frame for surveys may have missing values. Some question have not been responded by some testers, there are some NA values within the dataset that need to be removed on treatment.
- Data frame for surveys is splitted according to Criterion factor to extend analyses.

3.2. Recommendations accuracy study

3.2.1. Exploratory analysis

Present results overview with basic statistics. Summarise basic statistics.

```
summary(accuracy)
```

```
##      Tester   Failure  Solution  Report      Accuracy
##  1      : 1   CNN:14   TBR:14   Step:28   Min.   :0.0000
##  4      : 1   TEM:14   ARR:14             1st Qu.:0.0000
##  5      : 1                               Median :1.0000
##  6      : 1                               Mean   :0.5714
##  7      : 1                               3rd Qu.:1.0000
##  8      : 1                               Max.   :1.0000
##  (Other):22
```

Analyse factors group average accuracy. Calculate mean and standard deviations per factor group (failure and solution).

```
# Calculate using group_by function from dplyr
accuracyStats <- group_by(accuracy, Failure, Solution) %>%
  summarise(count=n(), mean = mean(Accuracy,na.rm = TRUE),
            sd = sd(Accuracy,na.rm = TRUE))
# Export tabulated result as csv
write.csv(x = accuracyStats, file = file.path("Tables", "7-AccuracyStats.csv"))
```

Graphically analyse variances per factors group (solution and operation). Plot accuracy percentages per solution as bar chart.

```
# Plot bars using abovementioned rationales and prepared theme
accuracyStatsPlot <- ggplot(accuracyStats, aes(x = Solution, y = mean, fill = Solution)) +
  geom_col() +
  geom_errorbar(aes(ymin = mean - sd/count, ymax = mean + sd/count),
                width = 0.5, colour = "gray60", lwd = 0.5) +
  geom_text(aes(label = scales::percent(mean)),
            stat = "identity", hjust = -0.5, vjust = -0.35, family = "Times", size = 3) +
  facet_grid(. ~ Failure) +
  scale_y_continuous(labels = scales::percent, limits = c(0,1)) +
  scale_fill_manual(values = c06Palette) +
  labs(y = "Accuracy percentage", title = "Failure") +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 11, hjust = 0.5)) +
  plotTheme
# Save plot
plot_save(accuracyStatsPlot, file.path("Figures", "5-AccuracyStatsPlot.png"), "png")
```

3.2.2. Variance analysis

Analyse significance of variances on accuracy results per solution. Calculate t-test for accuracy results per solution.

```
# Run t-test according to abovementioned response and interactions.
accuracyTT <- t.test(Accuracy ~ Solution, data = accuracy)
# Export tabulated results as csv
capture.output(accuracyTT, file = file.path("Tables", "8-AccuracyTT.csv"))
```

3.3. Reporting errors study

3.3.1. Correlation analysis

Evaluate correlation between errors and seconds response variables. Pearson null hypothesis: "There is no statistically significant relationship between variables".

```
# Create to analyse errors and seconds correlation
se <- na.omit(data.frame(Seconds = seconds$Seconds, Errors = errors$Percent))
# Evaluate correlation coefficient using Pearson's method
seCR <- cor.test(se$Seconds, se$Errors, method = "pearson", na.action = na.omit)
# Export tabulated results as csv
capture.output(seCR, file = file.path("Tables", "12-SECR.csv"))
```

Plot correlation between response variables errors and seconds. Cohen's interpretation: "Effect size = {(0.1, Small), (0.3, Moderate), (0.5, Large)}".

```
# Plot points and line using ggscatter from ggpubr
seCRPlot <- ggscatter(se, x = "Seconds", y = "Errors",
                      add = "reg.line", conf.int = TRUE,
                      cor.coef = TRUE, cor.method = "pearson") +
  scale_y_continuous(labels = percent, limits = c(0,1)) +
  labs (y = "Errors rates") +
  plotTheme
# Save plot in an independent file
ggsave(filename = file.path("Figures","9-SECRPlot.png"),
       plot = seCRPlot, device = "png",
       width = 15.9, height = 9.9375, units = "cm", dpi = 500)
```

3.3.2. Exploratory analysis

Present results overview with basic statistics. Summarise basic statistics.

```
summary(errors)
```

```
##      Tester    Failure  Solution            Report       Errors
##  1      :  6   CNN:84   TBN:42   DiagnosesState:56   Min.   :0.000
##  4      :  6   TEM:84   TBR:42   EvaluatesState:56   1st Qu.:0.000
##  5      :  6            ARN:42   Step          :56   Median :1.000
##  6      :  6            ARR:42                       Mean   :1.164
##  7      :  6                                         3rd Qu.:2.000
##  8      :  6                                         Max.   :5.000
##  (Other):132                                         NA's   :28
##     Correct          Total        Percent
##  Min.   :2.000   Min.   :7.0   Min.   :0.0000
##  1st Qu.:6.000   1st Qu.:7.0   1st Qu.:0.0000
##  Median :7.000   Median :8.0   Median :0.1300
##  Mean   :6.436   Mean   :7.6   Mean   :0.1581
##  3rd Qu.:7.000   3rd Qu.:8.0   3rd Qu.:0.2500
##  Max.   :8.000   Max.   :8.0   Max.   :0.7100
##  NA's   :28      NA's   :28    NA's   :28
```

Analyse factors group average errors Calculate mean and standard deviations per factor group (failure and solution).

```
# Calculate using group_by function from dplyr
errorsStats <- group_by(errors, Failure, Solution) %>%
  summarise(count=n(), mean = mean(Percent,na.rm = TRUE),
            sd = sd(Percent,na.rm = TRUE))
# Export tabulated result as csv
write.csv(x = errorsStats, file = file.path("Tables", "9-ErrorsStats.csv"))
```

Graphically analyse variances per factors group (failure and solution). Plot average errors per test as box and whiskers plot per failure and solution.

```
# Plot box and whiskers using abovementioned rationales and prepared theme
errorsStatsPlot <- ggplot(errors, aes(x = Solution, y = Percent, fill = Solution)) +
  geom_boxplot(colour = "gray60", lwd = 0.5, fatten = 2, na.rm = TRUE) +
  facet_grid(. ~ Failure) +
  scale_y_continuous(labels = percent, limits = c(0,1)) +
  scale_fill_manual(values = c06Palette) +
  labs(y = "Percentage of errors per reporting task", title = "Failure") +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 11, hjust = 0.5)) +
  plotTheme
# Save plot
plot_save(errorsStatsPlot, file.path("Figures", "6-ErrorsStatsPlot.png"), "png")
```

3.3.3. Variance analysis

3.3.3.1. Assumptions testing: normality, linearity, homogeneity

Prepare data for in-depth analysis by removing outliers. Use subset function with boxplot stats to manually identify and remove outliers. Fit linear model and calculate residuals and predictors.

```
# Use box stats to locate and remove outliers
subset(errors,errors$Percent %in% boxplot.stats(errors$Percent)$out)
```

```
##    Tester Failure Solution Report Errors Correct Total Percent
## 7       4     TEM      TBN   Step      5       2     7    0.71
## 94     18     TEM      TBR   Step      5       2     7    0.71
```

```
errorsClean <- na.omit(subset(errors, !(errors$Tester %in% c(4,18))))
# Fit data into linear model according to relevant factors
errorsLM <- lm(Percent ~ Failure*Solution, data = errorsClean)
# Calculate residuals, predicted values and squared predicted values
errorsClean$Residuals <- residuals(errorsLM)
errorsClean$Predicted <- predict(errorsLM)
errorsClean$SqrdPredicted <- predict(errorsLM)^2
```

Graphically test normallity plotting histogram of residuals. Plot residuals and normal distribution for graphical testing.

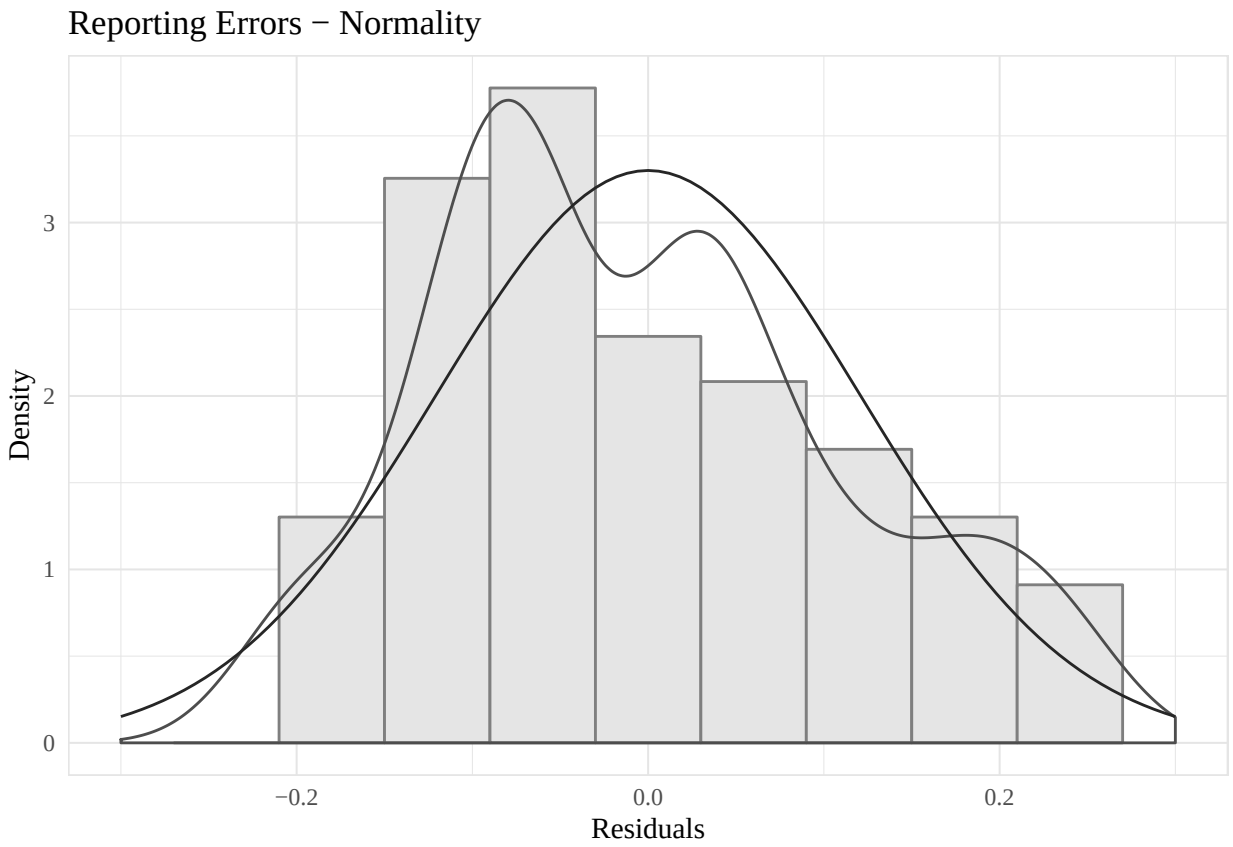
```
# Plot residuals histogram and overlay normal distribution using prepared theme
errorsNMPlot <- ggplot(errorsClean) +
  geom_histogram(aes(x = Residuals, y = ..density..), binwidth = 0.06,
                 fill = "gray90", colour = "grey50") +
  geom_density(aes(x = Residuals, y = ..density..), colour = "grey30") +
  stat_function(fun = function(x,mean,sd,n){
    dnorm(x = x, mean = mean, sd = sd)
  }, args = with(errorsClean, c(mean = mean(Residuals), sd = sd(Residuals),
```

```
                                  n = length(Residuals))), colour = "grey15") +
  scale_x_continuous(limits = c(-0.3,0.3)) +
  labs(title = "Reporting Errors - Normality", y = "Density") +
  plotTheme
# Visualise plot
errorsNMPlot
```

```
## Warning: Removed 2 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```




```
# Save plot
plot_save(errorsNMPlot, file.path("Figures", "7-ErrorsNMPlot.png"), "png")
```

```
## Warning: Removed 2 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing non-finite values (stat_density).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

Test normality with shapiro test. Null hypothesis: "Population is normally distributed". Reject null hypothesis with a significance threshold of p-value < 0.05.

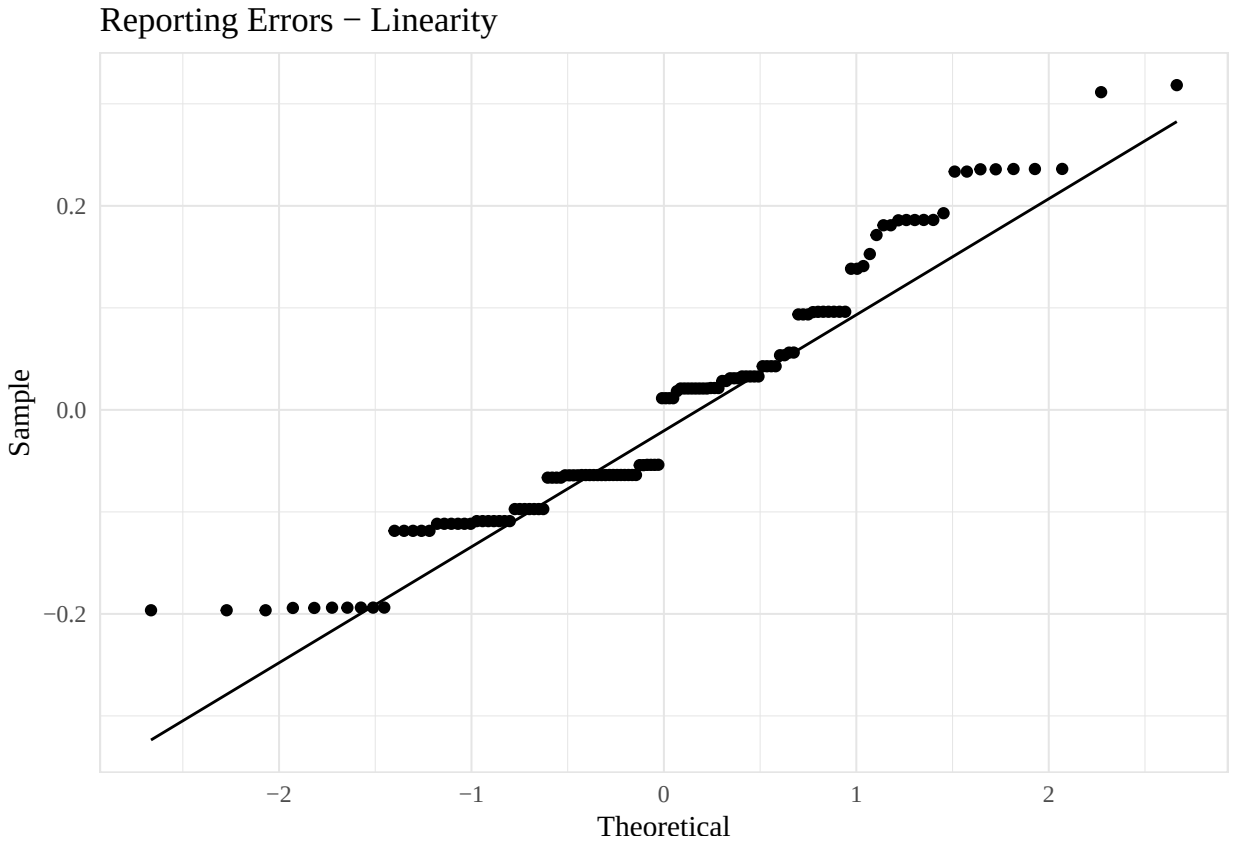
```
# Run shapiro test for normality
shapiro.test(errorsClean$Residuals)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  errorsClean$Residuals
## W = 0.94738, p-value = 7.172e-05
```

Graphically test linearity plotting a predicted quantiles versus sample quantiles. Plot residuals and samples and check against diagonal for graphical testing.

```
# Plot residuals and samples using qqplot and prepared theme
errorsQQPlot <- ggplot(errorsClean, aes(sample = Residuals)) +
  stat_qq() + stat_qq_line() +
  labs(title = "Reporting Errors - Linearity", x = "Theoretical", y = "Sample") +
  plotTheme
# Visualise plot
errorsQQPlot
```




```
# Save plot
plot_save(errorsQQPlot, file.path("Figures", "8-ErrorsQQPlot.png"), "png")
```

Test homogeneity assumption with Bartlett test. Null hypothesis: "All k population variances are equal" Reject null hypothesis with a significance threshold of p-value < 0.05.

```
bartlett.test(Percent ~ interaction(Failure, Solution), data = errorsClean)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  Percent by interaction(Failure, Solution)
## Bartlett's K-squared = 7.5683, df = 7, p-value = 0.3722
```

3.3.3.2. ANOVA and TukeyHSD tests

Analyse significance of variances on errors results per failure and solution. Calculate two-way ANOVA for errors results per failure and solution.

```
# Run anova test according to abovementioned response and interactions.
errorsAOV <- aov(Percent ~ Failure*Solution, data = errorsClean, na.action = na.omit)
# Export tabulated results as csv
capture.output(summary(errorsAOV), file = file.path("Tables", "10-ErrorsAOV.csv"))
```

Test differences between factor groups means using Tukey HSD test. Reject null hypotheses with a significance threshold of p-adj-value < 0.05.

```
# Run post-hoc pairwise t-test comparisons using TukeyHSD function.
errorsTHSD <- TukeyHSD(aov(Percent ~ Solution, data = errorsClean, na.action = na.omit))
# Export tabulated results as csv
capture.output(errorsTHSD, file = file.path("Tables", "11-ErrorsTHSD.csv"))
```

3.4. Reporting time study

3.4.1. Exploratory analysis

Present results overview with basic statistics. Summarise basic statistics.

```
summary(seconds)
```

```
##      Tester    Failure  Solution            Report      Seconds
##  1      :  6   CNN:84   TBN:42   DiagnosesState:56   Min.   : 78
##  4      :  6   TEM:84   TBR:42   EvaluatesState:56   1st Qu.:154
##  5      :  6            ARN:42   Step          :56   Median :184
##  6      :  6            ARR:42                       Mean   :184
##  7      :  6                                         3rd Qu.:199
##  8      :  6                                         Max.   :553
##  (Other):132                                         NA's   :28
```

Analyse factors group average seconds. Calculate mean and standard deviations per factor group (failure and solution).

```
# Calculate using group_by function from dplyr
secondsStats <- group_by(seconds, Failure, Solution) %>%
  summarise(count=n(), mean = mean(Seconds,na.rm = TRUE),
            sd = sd(Seconds,na.rm = TRUE))
# Export tabulated result as csv
write.csv(x = secondsStats, file = file.path("Tables", "13-SecondsStats.csv"))
```

Graphically analyse variances per factors group (failure and solution). Plot average seconds per test as box and whiskers plot per failure and solution.

```
# Plot box and whiskers using abovementioned rationales and prepared theme
secondsStatsPlot <- ggplot(seconds, aes(x = Solution, y = Seconds, fill = Solution)) +
  geom_boxplot(colour = "gray60", lwd = 0.5, fatten = 2, na.rm = TRUE) +
  facet_grid(. ~ Failure) +
  scale_y_continuous(limits = c(0,300)) +
  scale_fill_manual(values = c06Palette) +
  labs(y = "Number of seconds per reporting experiment", title = "Failure") +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 11, hjust = 0.5)) +
  plotTheme
```

```
# Save plot
plot_save(secondsStatsPlot, file.path("Figures", "10-SecondsStatsPlot.png"), "png")
```

3.4.2. Variance analysis

3.4.2.1. Assumptions testing: normality, linearity, homogeneity

Prepare data for in-depth analysis by removing outliers. Use subset function with boxplot stats to manually identify and remove outliers. Fit linear model and calculate residuals and predictors.

```
# Use box stats to locate and remove outliers
subset(seconds,seconds$Seconds %in% boxplot.stats(seconds$Seconds)$out)
```

```
##    Tester Failure Solution         Report Seconds
## 2       1     TEM      ARR EvaluatesState      86
## 17      5     CNN      TBR EvaluatesState     283
## 53     11     CNN      TBN EvaluatesState     277
## 55     12     TEM      TBN           Step     553
## 56     12     TEM      TBN EvaluatesState     383
## 57     12     TEM      TBN DiagnosesState     274
## 79     16     CNN      ARR           Step      79
## 89     17     TEM      ARN EvaluatesState     297
## 97     19     TEM      ARR           Step      78
```

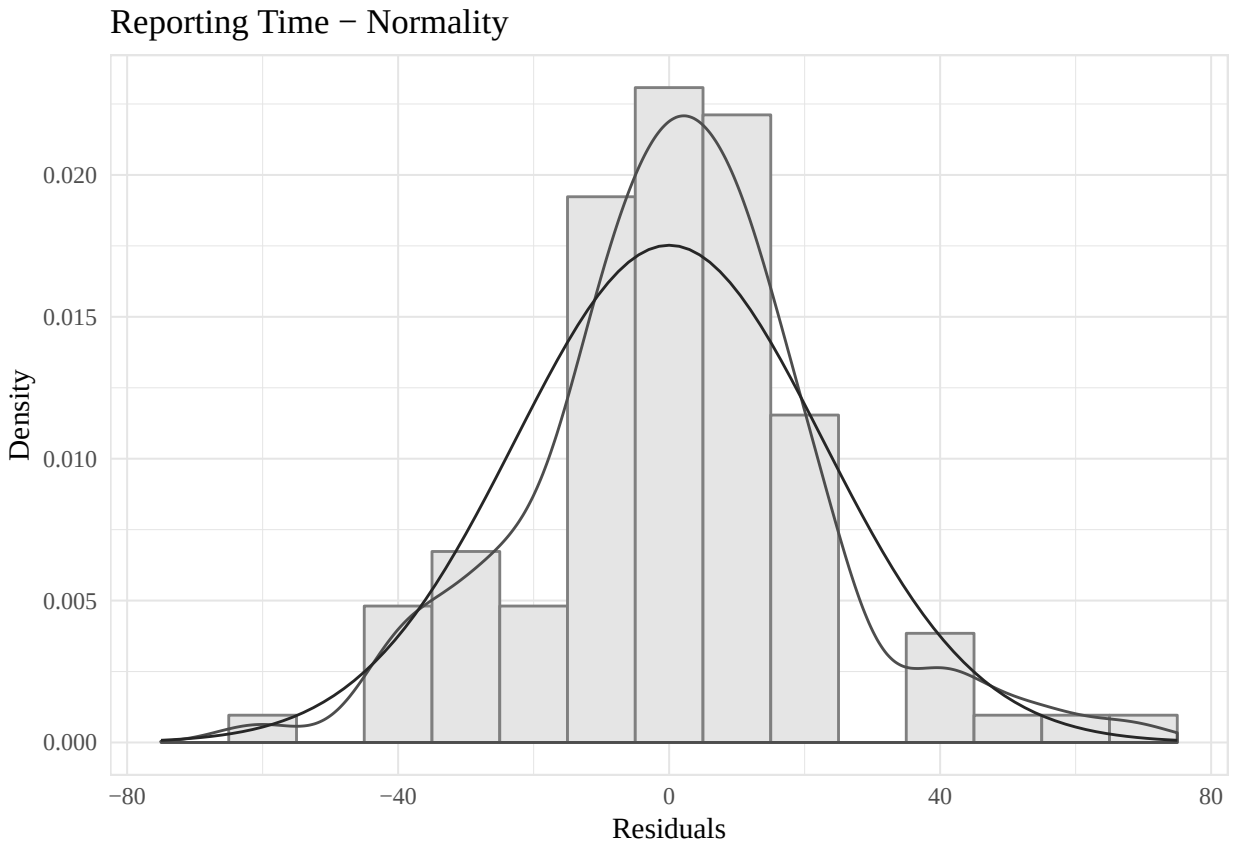
```
secondsClean <- na.omit(subset(seconds, !(seconds$Tester %in% c(1,5,11,12,16,17,19))))
# Fit data into linear model according to relevant factors
secondsLM <- lm(Seconds ~ Failure*Solution, data = secondsClean)
# Calculate residuals, predicted values and squared predicted values
secondsClean$Residuals <- residuals(secondsLM)
secondsClean$Predicted <- predict(secondsLM)
secondsClean$SqrdPredicted <- predict(secondsLM)^2
```

Graphically test normallity plotting histogram of residuals. Plot residuals and normal distribution for graphical testing.

```
# Plot residuals histogram and overlay normal distribution using prepared theme
secondsNMPlot <- ggplot(secondsClean) +
  geom_histogram(aes(x = Residuals, y = ..density..), binwidth = 10,
                 fill = "gray90", colour = "grey50") +
  geom_density(aes(x = Residuals, y = ..density..), colour = "grey30") +
  stat_function(fun = function(x,mean,sd,n){
    dnorm(x = x, mean = mean, sd = sd)
  }, args = with(secondsClean, c(mean = mean(Residuals), sd = sd(Residuals),
                                 n = length(Residuals))), colour = "grey15") +
  scale_x_continuous(limits = c(-75,75)) +
  labs(title = "Reporting Time - Normality", y = "Density") +
  plotTheme
# Visualise plot
secondsNMPlot
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```




```
# Save plot
plot_save(secondsNMPlot, file.path("Figures", "11-SecondsNMPlot.png"), "png")
```

```
## Warning: Removed 1 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 1 rows containing non-finite values (stat_density).
```

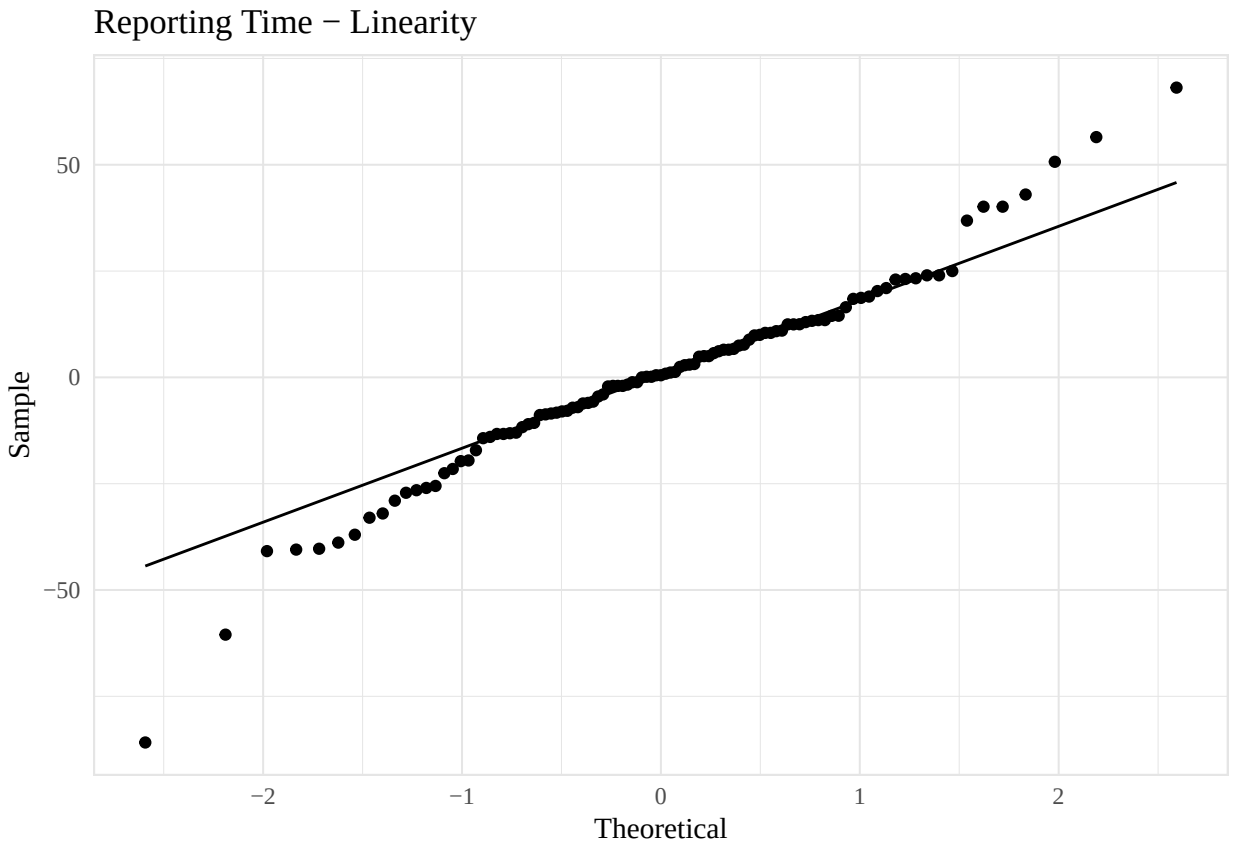
Test normality with shapiro test. Null hypothesis: "Population is normally distributed". Reject null hypothesis with a significance threshold of p-value < 0.05.

```
# Run shapiro test for normality
shapiro.test(secondsClean$Residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  secondsClean$Residuals
## W = 0.96666, p-value = 0.009562
```

Graphically test linearity plotting a predicted quantiles versus sample quantiles. Plot residuals and samples and check against diagonal for graphical testing.

```
# Plot residuals and samples using qqplot and prepared theme
secondsQQPlot <- ggplot(secondsClean, aes(sample = Residuals)) +
  stat_qq() + stat_qq_line() +
  labs(title = "Reporting Time - Linearity", x = "Theoretical", y = "Sample") +
  plotTheme
# Visualise plot
secondsQQPlot
```




```
# Save plot
plot_save(secondsQQPlot, file.path("Figures", "12-SecondsQQPlot.png"), "png")
```

Test homogeneity assumption with Bartlett test. Null hypothesis: "All k population variances are equal" Reject null hypothesis with a significance threshold of p-value < 0.05.

```
bartlett.test(Seconds ~ interaction(Failure, Solution), data = secondsClean)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  Seconds by interaction(Failure, Solution)
## Bartlett's K-squared = 22.081, df = 7, p-value = 0.00246
```

3.4.2.2. ANOVA and TukeyHSD tests

Analyse significance of variances on seconds results per failure and solution. Calculate two-way ANOVA for errors seconds per failure and solution.

```
# Run anova test according to abovementioned response and interactions.
secondsAOV <- aov(Seconds ~ Failure*Solution, data = seconds, na.action = na.omit)
# Export tabulated results as csv
capture.output(summary(secondsAOV), file = file.path("Tables", "14-SecondsAOV.csv"))
```

Test differences between factor groups means using Tukey HSD test. Reject null hypotheses with a significance threshold of p-adj-value < 0.05.

```
# Run post-hoc pairwise t-test comparisons using TukeyHSD function.
secondsTHSD <- TukeyHSD(aov(Seconds ~ Failure:Solution, data = seconds, na.action = na.omit))
```

```
# Export tabulated results as csv
capture.output(secondsTHSD, file = file.path("Tables", "15-SecondsTHSD.csv"))
```

3.5. Reporting workload study

Present results overview with basic statistics. Summarise basic statistics.

```
summary(workload)
```

```
##      Tester              Criterion      Weight       RawRate
##  1      :  6   Mental Demand  :28   Min.   :0.0   Min.   : 0.000
##  4      :  6   Physical Demand:28   1st Qu.:1.0   1st Qu.: 4.000
##  5      :  6   Temporal Demand:28   Median :3.0   Median :10.000
##  6      :  6   Performance    :28   Mean   :2.5   Mean   : 9.232
##  7      :  6   Effort         :28   3rd Qu.:4.0   3rd Qu.:14.000
##  8      :  6   Frustration    :28   Max.   :5.0   Max.   :20.000
##  (Other):132
##   AdjustedRate     WeightedRate
##  Min.   :  0.00   Min.   :0.000
##  1st Qu.:  8.00   1st Qu.:0.533
##  Median : 20.00   Median :1.333
##  Mean   : 26.15   Mean   :1.743
##  3rd Qu.: 40.00   3rd Qu.:2.667
##  Max.   :100.00   Max.   :6.667
##
```

Analyse criterions averages weights. Calculate weight mean and standard deviations per criterion.

```
# Calculate using group_by function from dplyr
wWeightsStats <- group_by(workload, Criterion) %>%
  summarise(count=n(), mean = mean(Weight,na.rm = TRUE),
            sd = sd(Weight,na.rm = TRUE))
# Export tabulated result as csv
write.csv(x = wWeightsStats, file = file.path("Tables", "16-WWeightsStats.csv"))
```

Graphically analyse weight variances per criterion. Plot average weights per tester as box and whiskers plot per criterion.

```
# Plot box and whiskers using abovementioned rationales and prepared theme
wWeightsStatsPlot <- ggplot(workload, aes(x = Criterion, y = Weight, fill = Criterion)) +
  geom_boxplot(colour = "gray60", lwd = 0.5, fatten = 2, na.rm = TRUE) +
  scale_y_continuous(limits = c(0,5)) +
  scale_fill_manual(values = c06Palette) +
  labs(y = "Weight") +
  theme(legend.position = "none",
        plot.title = element_text(size = 11, hjust = 0.5),
        axis.text.x = element_text(size = 8)) +
  plotTheme
# Save plot
plot_save(wWeightsStatsPlot, file.path("Figures", "13-WWeightsStatsPlot.png"), "png")
```

Analyse criterions averages raw rates. Calculate raw rates mean and standard deviations per criterion.

```
# Calculate using group_by function from dplyr
wRawRateStats <- group_by(workload, Criterion) %>%
  summarise(count=n(), mean = mean(RawRate,na.rm = TRUE),
```

```
            sd = sd(RawRate,na.rm = TRUE))
# Export tabulated result as csv
write.csv(x = wRawRateStats, file = file.path("Tables", "17-WRawRatesStats.csv"))
```

Graphically analyse raw rate variances per criterion. Plot average raw rates per tester as box and whiskers plot per criterion.

```
# Plot box and whiskers using abovementioned rationales and prepared theme
wRawRatesStatsPlot <- ggplot(workload, aes(x = Criterion, y = RawRate, fill = Criterion)) +
  geom_boxplot(colour = "gray60", lwd = 0.5, fatten = 2, na.rm = TRUE) +
  scale_y_continuous(limits = c(0,20)) +
  scale_fill_manual(values = c06Palette) +
  labs(y = "Raw Rate") +
  theme(legend.position = "none",
        plot.title = element_text(size = 11, hjust = 0.5),
        axis.text.x = element_text(size = 8)) +
  plotTheme
# Save plot
plot_save(wRawRatesStatsPlot, file.path("Figures", "14-WRawRatesStatsPlot.png"), "png")
```

Analyse criterions averages weighted rates. Calculate weighted rates mean and standard deviations per criterion.

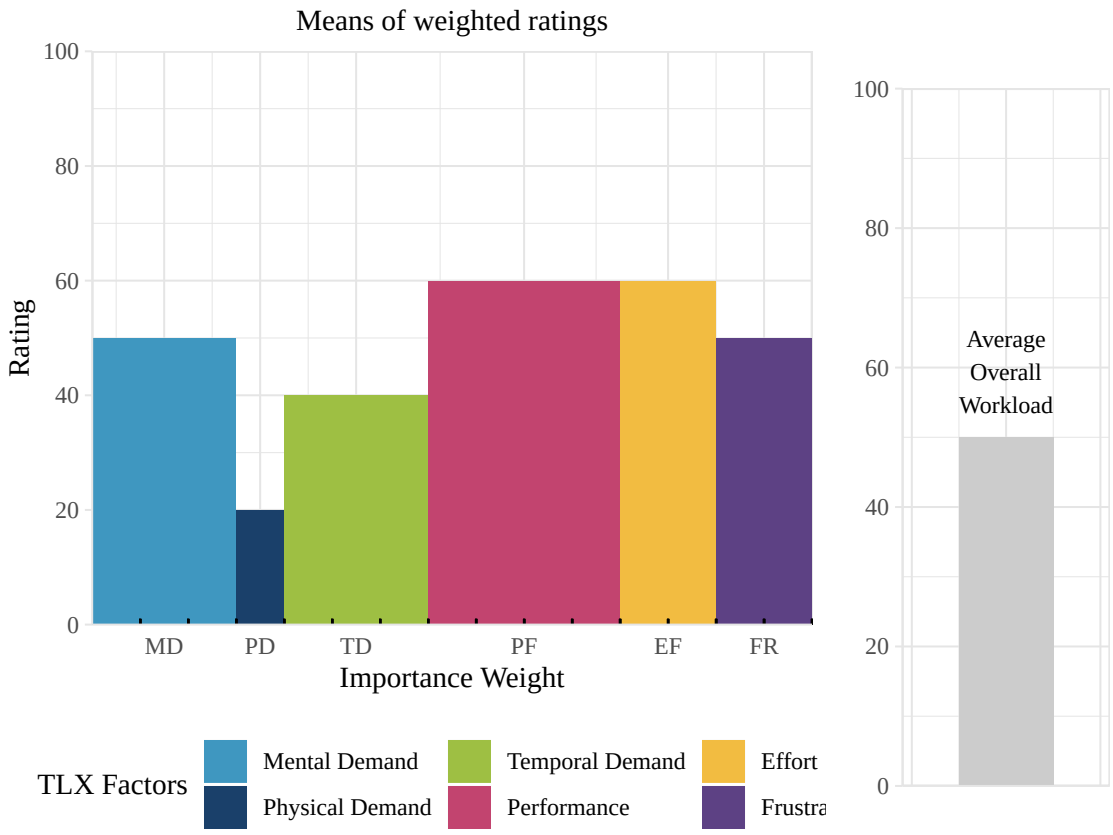
```
# Calculate using group_by function from dplyr
wWeightedRatesStats <- group_by(workload, Criterion) %>%
  summarise(count=n(), mean = mean(WeightedRate,na.rm = TRUE),
            sd = sd(WeightedRate,na.rm = TRUE))
# Export tabulated result as csv
write.csv(x = wWeightedRatesStats, file = file.path("Tables", "18-WWeightedRatesStats.csv"))
```

Graphically analyse weighted rate variances per criterion. Plot average weighted rates per tester as box and whiskers plot per criterion.

```
# Plot box and whiskers using abovementioned rationales and prepared theme
wWeightedRatesStatsPlot <- ggplot(workload, aes(x = Criterion, y = WeightedRate, fill = Criterion)) +
  geom_boxplot(colour = "gray60", lwd = 0.5, fatten = 2, na.rm = TRUE) +
  scale_y_continuous(limits = c(0,7)) +
  scale_fill_manual(values = c06Palette) +
  labs(y = "Weighted Rate") +
  theme(legend.position = "none",
        plot.title = element_text(size = 11, hjust = 0.5),
        axis.text.x = element_text(size = 8)) +
  plotTheme
# Save plot
plot_save(wWeightedRatesStatsPlot, file.path("Figures", "15-WWeightedRatesStatsPlot.png"), "png")
```

Graphically present overall weighted rates per criterion. Plot rates versus weights per criterion using Reski's function.

```
# Format average weights and raw rates
weightsF <- round_preserve_sum(round(wWeightsStats$mean),15)
rawRatesF <- 5*(2*round(wRawRateStats$mean/2))
workloadF <- data.frame(weight = weightsF, rawRating = rawRatesF)
# Produce tlx individual results
workloadTLX <- tlx.individual(workloadF)
```




```
# Save plots
plot_save(workloadTLX$factorPlot, file.path("Figures", "16-NASATLXFactorsPlot.png"), "png")
plot_save(workloadTLX$combinedPlot, file.path("Figures", "17-NASATLXWorkloadPlot.png"), "png")
```

3.6. Usability study

Present results overview with basic statistics. Summarise data structure and basic statistics.

```
summary(usability)
```

```
##      Tester    Solution          Criterion                   Aspect
##  1       : 28   TBN:196   Ease-To-Learn:168   Buttons-Gestures   : 56
##  4       : 28   TBR:196   Ease-To-Use  :168   Confidence-Increase: 56
##  5       : 28   ARN:196   Accuracy     :  0   Design             : 56
##  6       : 28   ARR:196   Effectiveness:280   Efficiency-Increase: 56
##  7       : 28             Satisfaction :168   End-Ease           : 56
##  8       : 28                                 Error-Reduction    : 56
##  (Other):616                                  (Other)            :448
##     Response
##  Min.   :1.000
##  1st Qu.:3.000
##  Median :4.000
##  Mean   :3.741
##  3rd Qu.:4.000
##  Max.   :5.000
##
```

Analyse average responses per criterion, solution and failure. Calculate mean and standard deviations per

factor group.

```
# Calculate using group_by function from dplyr
usabilityStats <- group_by(usability, Criterion, Solution) %>%
  summarise(count=n(), mean = mean(Response,na.rm = TRUE),
            sd = sd(Response,na.rm = TRUE))
# Export tabulated result as csv
write.csv(x = usabilityStats, file = file.path("Tables", "19-UsabilityStats.csv"))
```

Graphically analyse responses averages for each criterion per solution. Plot average responses count per tester as box and whiskers per criterion and solution with conservative average for Likert scale.

```
# Plot bars using na omitted responses, abovementioned rationale and prepared theme
usabilityStatsPlot <- ggplot(na.omit(usability), aes(x = Criterion, y = Response, fill = Solution)) +
  geom_boxplot(colour = "gray60", lwd = 0.5, fatten = 2, na.action = na.omit) +
  geom_hline(yintercept = 3.5, linetype = "dashed", color = "gray60") +
  scale_y_continuous(limits = c(1,5)) +
  scale_fill_manual(values = c06Palette) +
  labs(y = "Likert scale") +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 11, hjust = 0.5)) +
  plotTheme
```

```
## Warning: Ignoring unknown parameters: na.action
```

```
# Save plot
plot_save(usabilityStatsPlot, file.path("Figures", "18-UsabilityStatsPlot.png"), "png")
```

Graphically analyse responses averages for each aspect regarding Ease-To-Learn criterion per solution. Plot average responses count per tester as box and whiskers per aspect and solution with conservative average for Likert scale.

```
# Plot bars using abovementioned rationale and prepared theme
usabilityELPlot <- ggplot(usabilityEL, aes(x = Aspect, y = Response, fill = Solution)) +
  geom_boxplot(colour = "gray60", lwd = 0.5, fatten = 2, na.action = na.omit) +
  geom_hline(yintercept = 3.5, linetype = "dashed", color = "gray60") +
  scale_y_continuous(limits = c(1,5)) +
  scale_fill_manual(values = c06Palette) +
  labs(y = "Likert scale") +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 11, hjust = 0.5)) +
  plotTheme
```

```
## Warning: Ignoring unknown parameters: na.action
```

```
# Save plot
plot_save(usabilityELPlot, file.path("Figures", "19-UsabilityELPlot.png"), "png")
```

Graphically analyse responses averages for each aspect regarding Ease-To-Use criterion per solution. Plot average responses count per tester as box and whiskers per aspect and solution with conservative average for Likert scale.

```
# Plot bars using abovementioned rationale and prepared theme
usabilityEUPlot <- ggplot(usabilityEU, aes(x = Aspect, y = Response, fill = Solution)) +
  geom_boxplot(colour = "gray60", lwd = 0.5, fatten = 2, na.action = na.omit) +
  geom_hline(yintercept = 3.5, linetype = "dashed", color = "gray60") +
  scale_y_continuous(limits = c(1,5)) +
  scale_fill_manual(values = c06Palette) +
```

```
  labs(y = "Likert scale") +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 11, hjust = 0.5)) +
  plotTheme
```

```
## Warning: Ignoring unknown parameters: na.action
```

```
# Save plot
plot_save(usabilityEUPlot, file.path("Figures", "20-UsabilityEUPlot.png"), "png")
```

Graphically analyse responses averages for each aspect regarding Effectiveness criterion per solution. Plot average responses count per tester as box and whiskers per aspect and solution with conservative average for Likert scale.

```
# Plot bars using abovementioned rationale and prepared theme
usabilityEFPlot <- ggplot(usabilityEF, aes(x = Aspect, y = Response, fill = Solution)) +
  geom_boxplot(colour = "gray60", lwd = 0.5, fatten = 2, na.action = na.omit) +
  geom_hline(yintercept = 3.5, linetype = "dashed", color = "gray60") +
  scale_y_continuous(limits = c(1,5)) +
  scale_fill_manual(values = c06Palette) +
  labs(y = "Likert scale") +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 11, hjust = 0.5),
        axis.text.x = element_text(size = 8)) +
  plotTheme
```

```
## Warning: Ignoring unknown parameters: na.action
```

```
# Save plot
plot_save(usabilityEFPlot, file.path("Figures", "21-UsabilityEFPlot.png"), "png")
```

Graphically analyse responses averages for each aspect regarding Satisfaction criterion per solution. Plot average responses count per tester as box and whiskers per aspect and solution with conservative average for Likert scale.

```
# Plot bars using abovementioned rationale and prepared theme
usabilitySTPlot <- ggplot(usabilityST, aes(x = Aspect, y = Response, fill = Solution)) +
  geom_boxplot(colour = "gray60", lwd = 0.5, fatten = 2, na.action = na.omit) +
  geom_hline(yintercept = 3.5, linetype = "dashed", color = "gray60") +
  scale_y_continuous(limits = c(1,5)) +
  scale_fill_manual(values = c06Palette) +
  labs(y = "Likert scale") +
  theme(legend.position = "bottom",
        plot.title = element_text(size = 11, hjust = 0.5)) +
  plotTheme
```

```
## Warning: Ignoring unknown parameters: na.action
```

```
# Save plot
plot_save(usabilitySTPlot, file.path("Figures", "22-UsabilitySTPlot.png"), "png")
```

Analyse average responses per aspect, criterion, solution and operation. Calculate mean and standard deviations per factor group.

```
# Calculate using group_by function from dplyr
usabilityAspectsStats <- group_by(usability, Criterion, Aspect, Solution) %>%
  summarise(count=n(), mean = mean(Response,na.rm = TRUE),
            sd = sd(Response,na.rm = TRUE))
```

```r
# Export tabulated result as csv
write.csv(x = usabilityAspectsStats, file = file.path("Tables", "20-UsabilityAspectsStats.csv"))
```

4. Results

This section aims to discuss experimental results obtained in **Section 3** regarding the validity of research hypotheses presented in **Section 2.2**.

4.1. Recommendations accuracy results

Stopwatch experiments consisted of testers completing diagnosis reporting operations regarding two different failures' root causes: electric (CNN) and electronic (TEM). Recommendations accuracy is defined as the number of times recommender methods suggested failures' root causes and these were reported by testers. Testers utilised diverse AR (ARR) and non-AR (TBR) recommender solutions to report these root causes. Due to the differences in recommendation algorithms, ARR solution was hypothesised to provide higher accuracy than TBR.

Figure 5 and Table 7 display average accuracy results per solution and experimental failure. These results suggest a considerable diference in accuracy between ARR and TBR for both, electric (CNN) and electronic (TEM) failures. In electric failure experiments, testers right recommendation selection rate (accuracy) was almost double with ARR (85%) than with TBR (43%). In electronic failure experiments, accuracy was 2.4 times better with ARR (71%) compared to TBR (29%). These differences can be considered statistically significant (p-value = 0.022) with a confidence interval of 95% (p-value < 0.05) according to t-tests results presented in Table 8.

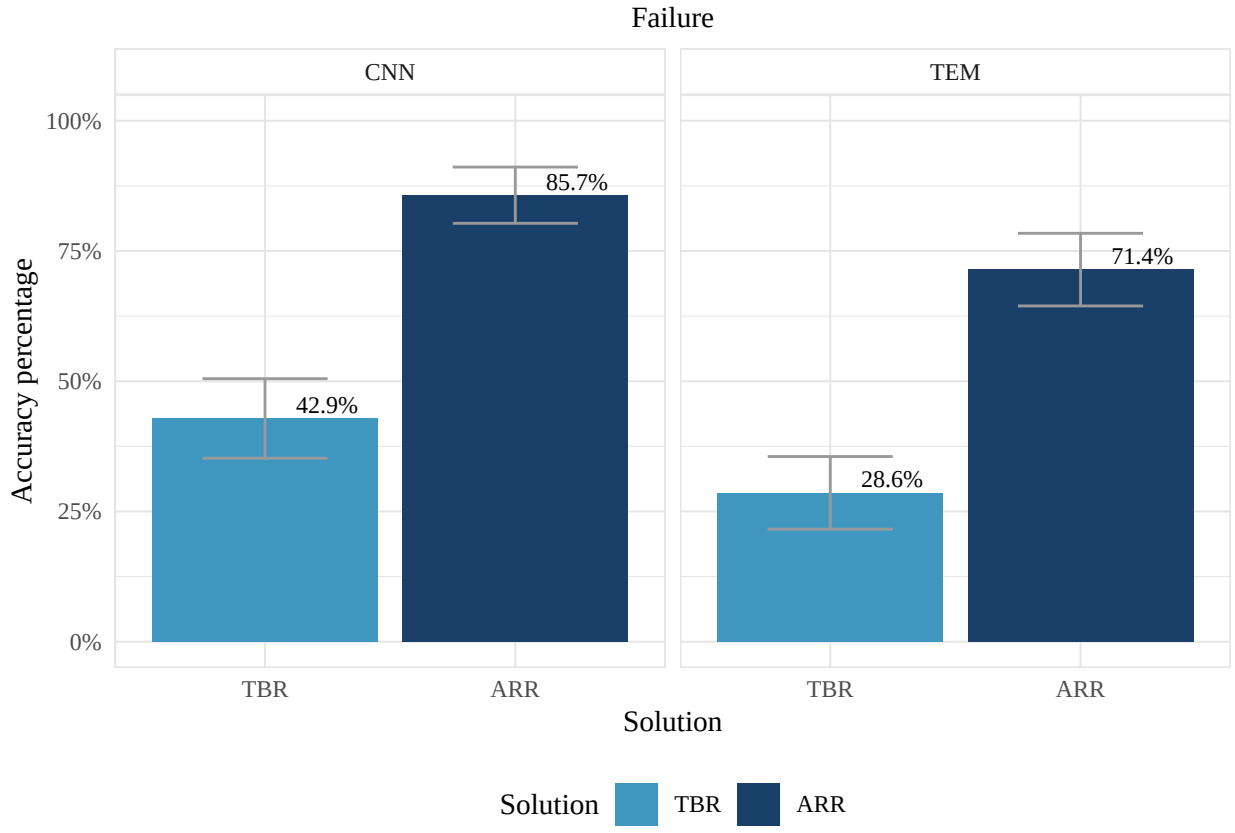


Figure 5: Box and whiskers plot on accuracy results per experimental solution and failure.

Overall, recommendations accuracy analyses indicate that the proposed AR-based content-aware recommender (ARR) has improved accuracy compared to more conventional ontology-based (TBR) recommendations.

Table 7: Means and std. deviations on recommendations accuracy per solution and failure factors.

Failure	Solution	count	mean	sd
CNN	TBR	7	0.4285714	0.5345225
CNN	ARR	7	0.8571429	0.3779645
TEM	TBR	7	0.2857143	0.4879500
TEM	ARR	7	0.7142857	0.4879500

Table 8: T-test results on recommendations accuracy variance per solution.

Welch.Two.Sample.t.test
data: Accuracy by Solution t = -2.4495, df = 25.399, p-value = 0.02153 alternative hypothesis: true difference in means is not equal to 0 95 percent confidence interval: -0.78862891 -0.06851395
sample estimates: mean in group TBR mean in group ARR 0.3571429 0.7857143

Based on the proposed solution's behaviour, there are two complementary explanations for these results. First, AR-based hybrid recommendations (ARR) are more precise and provide correct suggestions more often than conventional ontology-based tools (TBR). Second, augmented content formats provide easier visualisation of recommended fault conditions allowing the tester to choose correctly more often. Future works can investigate the independent effect of each cause more in-depth through experimentation in real-life conditions.

4.2. Reporting errors results

Stopwatch experiments also counted reporting errors, which aimed to measure reporting effectiveness through the number of mistakes in data input tasks. The number of data input tasks varies with the ontology class being instantiated. Hence, errors analyses evaluate percentage of errors by total number of data input tasks per experimental reporting task. According to validation hypotheses, errors rates are expected to decrease with the use of AR (ARN) and recommender (ARR, TBR) reporting solutions compared to alternative options (TBN).

Figure 6 and Table 9 display average errors rates per experimental solution and failure. These results indicate a considerable difference between AR (ARN, ARR) and non-AR (TBN, TBR) reporting solutions but no relevant effect of recommendations (ARR vs ARN, and TBR vs TBN) in errors reduction. For both experimental failures, average errors rates vary similarly per solution. Non-AR reporting tools had errors rates ranging between 19%-21%, while AR-based reporting methods achieved smaller errors rates ranging between 10%-12%. Besides, recommender solutions (ARR, TBR) had slightly higher errors rates compared to their non-recommender counterparts (ARN, TBN). This can be caused due to the impact of recommendations on data input tasks, which get reduced with the use of recommendations.

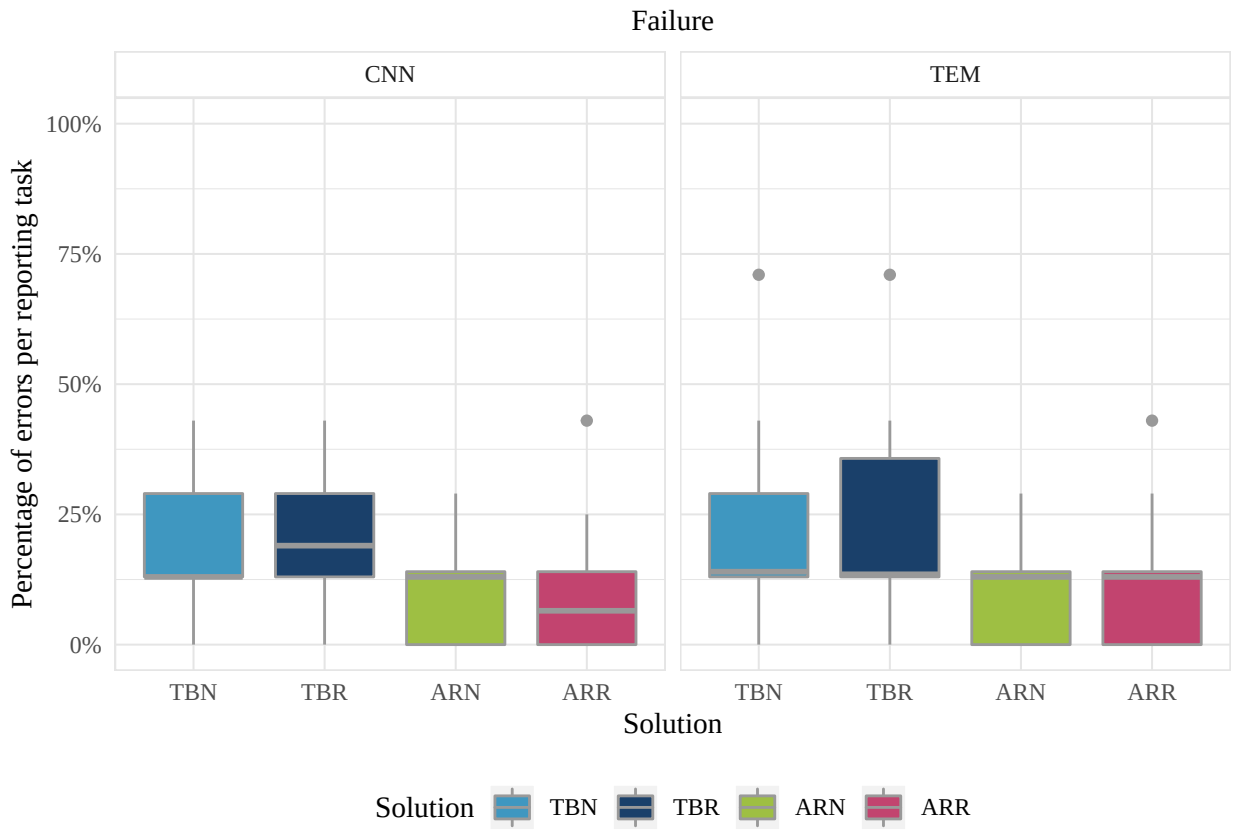


Figure 6: Box and whiskers plot on errors rates per experimental failure and solution.

Table 9: Means and std. deviations on errors rates per solution and failure factors.

Failure	Solution	count	mean	sd
CNN	TBN	21	0.1938095	0.1270227
CNN	TBR	21	0.1964286	0.1453170
CNN	ARN	21	0.1023810	0.0844929
CNN	ARR	21	0.1050000	0.1325925
TEM	TBN	21	0.2180952	0.1683633
TEM	TBR	21	0.2264286	0.1977414
TEM	ARN	21	0.1090476	0.0935363
TEM	ARR	21	0.1185714	0.1235910

Further analyses can identify the significance of differences discussed above. Table 10 presents a two-way ANOVA test conducted to analyse errors rates variance with failure and solution experimental factors. According to its results, it can be said with a confidence interval of 95% (p-value < 0.05), that solution is a significant factor (p-value $= 0.0019$) while failure and their interaction (solution:failure) are not. Moreover, post hoc comparisons results from the Tukey HSD test (Table 11) can help to evaluate differences between solutions on errors rates results. Comparisons results show that ARR and ARN errors rates are significantly different (p-value < 0.05) to TBR and TBN. They also show no significant differences between ARR and ARN, and TBR and TBN.

This research's hypotheses aimed for reporting errors and time to measure reporting effectiveness and efficiency, respectively. Nevertheless, these hypotheses assumed that errors and time are response variables that are not correlated. Pearson's correlation test [**ref**] can help to evaluate such correlation. Table 18 and Figure

Table 10: Two-way ANOVA test results on errors rates for failure and solution factors.

Df.Sum.Sq.Mean.Sq.F.value.Pr..F.
Failure 1 0.0002 0.00015 0.010 0.9215
Solution 3 0.2439 0.08131 5.263 0.0019 **
Failure:Solution 3 0.0010 0.00035 0.022 0.9954
Residuals 122 1.8848 0.01545
—
Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Table 11: Significance (p-value) results on post hoc comparisons (Tukey HSD) between solution and failure factors groups in errors rates results

Tukey.multiple.comparisons.of.means
95% family-wise confidence level
Fit: aov(formula = Percent ~ Solution, data = errorsClean, na.action = na.omit)
$Solution
diff lwr upr p adj
TBR-TBN 0.001538462 -0.07912274 0.082199667 0.9999558
ARN-TBN -0.090256410 -0.16240199 -0.018110835 0.0077923
ARR-TBN -0.078461538 -0.15912274 0.002199667 0.0597559
ARN-TBR -0.091794872 -0.17245608 -0.011133666 0.0188599
ARR-TBR -0.080000000 -0.16835992 0.008359924 0.0908473
ARR-ARN 0.011794872 -0.06886633 0.092456078 0.9811332

7 present its results. With a confidence interval of 95% (p-value < 0.05), it can be said that correlation between these two variables is not significant. Besides, the correlation can be classified as small ($r = 0.054$) according to Cohen's interpretation **[ref]**. Therefore, it can be said valid the assumption regarding errors as effectiveness measure and time as efficiency measure.

Table 12: Pearson's correlation test on reporting errors rates and times.

Pearson.s.product.moment.correlation
data: se$Seconds and se$Errors
t = 0.63101, df = 138, p-value = 0.5291
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
-0.1132736 0.2176046
sample estimates:
cor
0.05363767

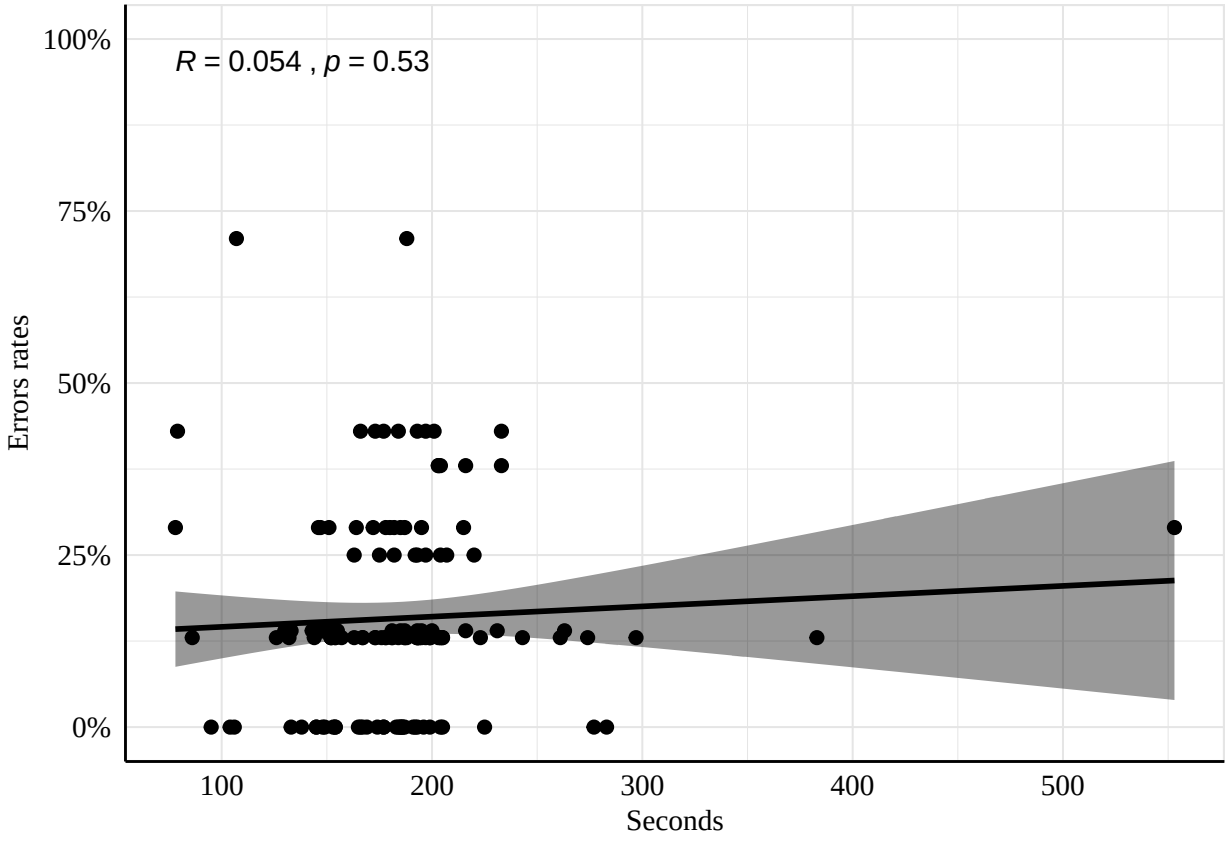


Figure 7: Scatter plot on reporting error rares and times correlation.

Overall, previous discussions support validity of the following considerations regarding the effect on completion errors of AR-based and recommendable reporting solutions:

- Correlation between reporting errors and times results cannot be considered significant.
- Errors can be considered a measure of reporting effectiveness and time a measure of reporting efficiency.
- Differences in errors rates among experimental failures cannot be considered statistically significant.
- Errors rates for AR reporting tools are half of those from non-AR reporting solutions.
- Differences in errors rates between AR and non-AR reporting solutions can be considered statistically significant.
- Errors rates for recommender tools are 10% higher than those from non-recommender reporting solutions.
- Differences in errors rates between recommender and non-recommender solutions cannot be considered statistically significant.
- Increase on errors rates for recommender tools can be caused due to the impact of recommendations in the number of data input tasks.

These results indicate the validity of some research's hypotheses regarding the effect of AR and recommenders in reporting effectiveness. AR content formats enabled more contextualise data input methods, which can be considered to have a positive effect on data input mistakes. Instead, recommenders reduce the number of data input tasks and so, their effect cannot be considered significant in reducing their errors.

4.3. Reporting time results

Stopwatch experiments lastly measured reporting time for evaluating reporting efficiency. Time is defined as the number of seconds taken by a tester to complete a diagnosis reporting task of a failur'e's root cause (CNN

and TEM). Testers employed diverse reporting solutions (ARR, ARN, TBR and TBN) to accomplish these tasks. Those solutions that implemented AR and recommendation methods were hypothesised to obtain faster reporting times than their counterparts.

Figure 8 and Table 13 present average reporting times per experimental solution and failures. These results indicate a considerable difference between AR and non-AR reporting solutions and AR recommendable and non-recommendable solutions. In electric failure experiments, results suggest that the proposed solution (ARR) reporting time is 20% faster than the AR non-recommender alternative and this is 6% faster than other non-AR alternatives. In electronic failure experiments, results show a similar difference between recommender and non-recommender AR solutions (22%) but a bigger difference between non-recommender AR and other non-AR solutions (TBN (20%) and TBR (6%)). Therefore, it can be said that both AR content and context-aware recommendations seem to have an effect on reporting time.

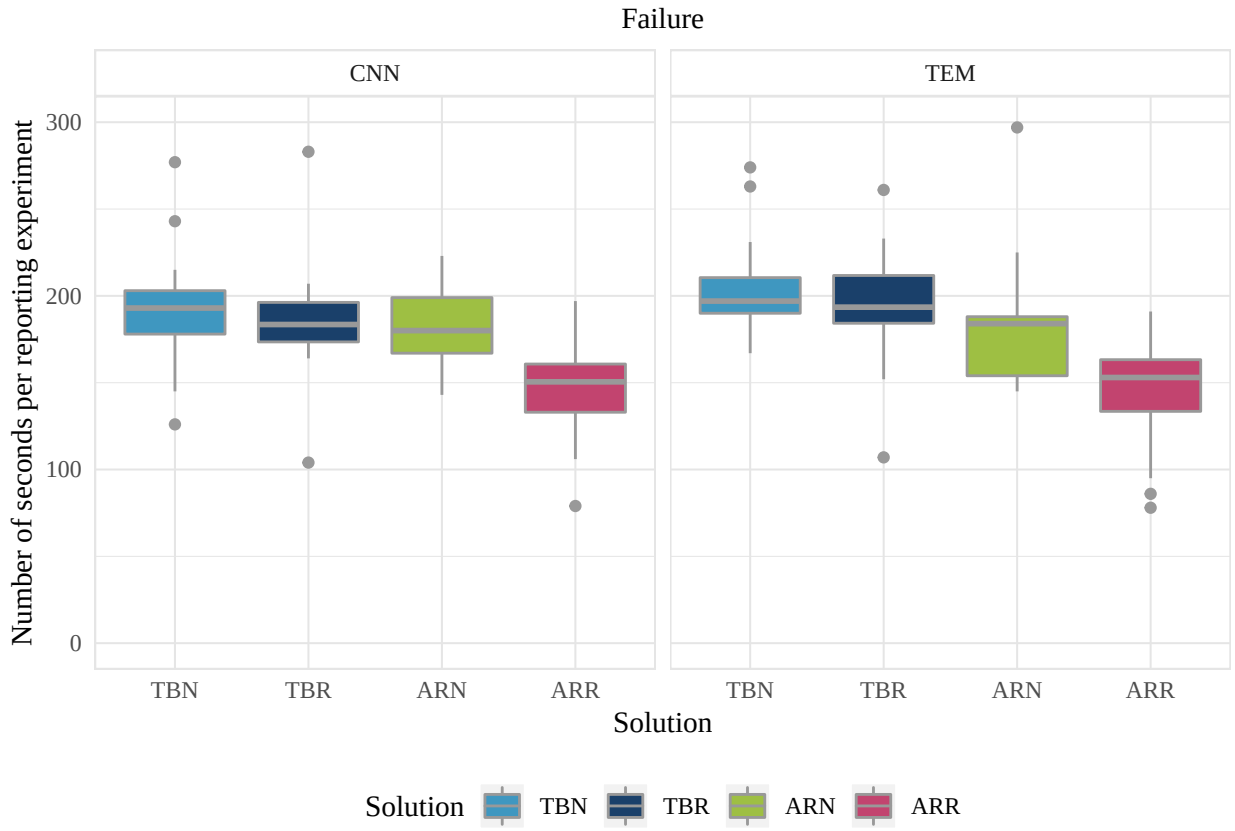


Figure 8: Box and whiskers plot on reporting times per experimental failure and solution.

Further analyses can identify the statistical significance of differences discussed above. Table 16 presents a two-way ANOVA test conducted to analyse reporting time variances with failure and solution experimental factors. According to its results, the solution factor effect is statistically significant (p-value = 4.54e-07) and the failure factor can also be considered significant (p-value = 0.08) but with a confidence interval smaller than 95% (p-value < 0.05). Moreover, post hoc comparisons results from the Tukey HSD test (Table 17) can help to evalute differences between solutions on reporting times per experimental failure. Comparison results indicate that ARR is significantly different to other alternatives in electric failure experiments, and significantly different to non-AR alternative solutions (TBN and TBR) in electronic failure experiments. Besides, the non-recommender AR solution (ARN) is not significantly different on reporting times achieved to non-AR alternatives in any experiment. Hence, it can be said that AR achieves improved reporting times when implementing AR-based context-aware recommendations.

Overall, previous discussions support validity of the following considerations regarding the effect on reporting

Table 13: Means and std. deviations on reporting times per solution and failure factors.

Failure	Solution	count	mean	sd
CNN	TBN	21	190.2857	32.57168
CNN	TBR	21	186.5000	37.39550
CNN	ARN	21	180.0000	21.95678
CNN	ARR	21	145.0714	29.14807
TEM	TBN	21	229.5714	87.56002
TEM	TBR	21	192.8571	38.37238
TEM	ARN	21	182.0000	33.80089
TEM	ARR	21	142.6429	33.97195

Table 14: Two-way ANOVA test results on errors rates for failure and solution factors.

Df.Sum.Sq.Mean.Sq.F.value...Pr..F.
Failure 1 6072 6072 2.964 0.0875 .
Solution 3 74638 24879 12.145 4.54e-07 ***
Failure:Solution 3 10500 3500 1.708 0.1684
Residuals 132 270415 2049
—
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
28 observations deleted due to missingness

time of AR-based and recommendable reporting solutions:

- Differences in times between experimental failures can be considered significant.
- Differences among AR-recommender (ARR) solutions and alternative solutions (ARN, TBR, TBN) can be considered significant.
- Differences among AR-non-recommender (ARR) solutions and alternative solutions (TBR, TBN) cannot be considered significant.
- In electric failure experiments, reporting time for ARR is 20%-23% faster than alternative solutions (ARN, TBR, TBN).
- In electronic failure experiments, reporting time for ARR is 22%-38% faster than alternative solutions (ARN, TBR, TBN).

These considerations indicate the validity of some research's hypotheses regarding the effect of AR and recommenders in reporting efficiency. AR content formats enabled significantly faster reporting times but only when implementing context-aware AR-based recommendations.

4.4. Reporting workload results

Workload surveys aim to evaluate testers perceived requisites on diagnosis reporting tasks for validating the effectiveness and efficiency measures assumption. Prior to experimentation, testers were asked to complete a pair-wise comparison survey for weighting workload factors described by NASA-TLX methodology **[ref]**. After experiments, testers were questioned to rate each workload factor in the experiments conducted. According to NASA-TLX workload factors (Table 6), testers were hypothesised to perceived experimental workload with higher requisites on Mental Demand and Performance for providing accurate reports on experimental failures' root causes.

Tables 16 and 17 and Figure 9 display testers' average responses regarding workload factors weights and scores. These results show that Performance and Effort were the most relevant factors as perceived by testers,

Table 15: Significance (p-value) results on post hoc comparisons (Tukey HSD) between solution and failure factors groups in errors rates results

Tukey.multiple.comparisons.of.means
95% family-wise confidence level
Fit: aov(formula = Seconds ~ Failure:Solution, data = seconds, na.action = na.omit)
$‘Failure:Solution‘
diff lwr upr p adj
TEM:TBN-CNN:TBN 39.285714 -3.737705 82.3091340 0.1006827
CNN:TBR-CNN:TBN -3.785714 -51.887360 44.3159313 0.9999974
TEM:TBR-CNN:TBN 2.571429 -45.530217 50.6730741 0.9999998
CNN:ARN-CNN:TBN -10.285714 -53.309134 32.7377054 0.9957076
TEM:ARN-CNN:TBN -8.285714 -51.309134 34.7377054 0.9989121
CNN:ARR-CNN:TBN -45.214286 -93.315931 2.8873598 0.0818081
TEM:ARR-CNN:TBN -47.642857 -95.744503 0.4587884 0.0542062
CNN:TBR-TEM:TBN -43.071429 -91.173074 5.0302170 0.1149377
TEM:TBR-TEM:TBN -36.714286 -84.815931 11.3873598 0.2745911
CNN:ARN-TEM:TBN -49.571429 -92.594848 -6.5480088 0.0122133
TEM:ARN-TEM:TBN -47.571429 -90.594848 -4.5480088 0.0192242
CNN:ARR-TEM:TBN -84.500000 -132.601646 -36.3983544 0.0000078
TEM:ARR-TEM:TBN -86.928571 -135.030217 -38.8269259 0.0000038
TEM:TBR-CNN:TBR 6.357143 -46.335570 59.0498555 0.9999516
CNN:ARN-CNN:TBR -6.500000 -54.601646 41.6016456 0.9998958
TEM:ARN-CNN:TBR -4.500000 -52.601646 43.6016456 0.9999915
CNN:ARR-CNN:TBR -41.428571 -94.121284 11.2641412 0.2397227
TEM:ARR-CNN:TBR -43.857143 -96.549856 8.8355698 0.1789045
CNN:ARN-TEM:TBR -12.857143 -60.958788 35.2445027 0.9915106
TEM:ARN-TEM:TBR -10.857143 -58.958788 37.2445027 0.9970027
CNN:ARR-TEM:TBR -47.785714 -100.478427 4.9069984 0.1055443
TEM:ARR-TEM:TBR -50.214286 -102.906998 2.4784269 0.0737799
TEM:ARN-CNN:ARN 2.000000 -41.023420 45.0234197 0.9999999
CNN:ARR-CNN:ARN -34.928571 -83.030217 13.1730741 0.3370781
TEM:ARR-CNN:ARN -37.357143 -85.458788 10.7445027 0.2539345
CNN:ARR-TEM:ARN -36.928571 -85.030217 11.1730741 0.2675937
TEM:ARR-TEM:ARN -39.357143 -87.458788 8.7445027 0.1962225
TEM:ARR-CNN:ARR -2.428571 -55.121284 50.2641412 0.9999999

with Mental Demand and Frustration following closely. Overall, these results suggest the validity of this research's hypothesis that enounces diagnosis reporting tasks as mentally and performance demanding.

Table 16: Means and std deviations on workload factors weights (scale 0-5) for diagnosis reporting tasks.

Criterion	count	mean	sd
Mental Demand	28	2.7500000	1.4304881
Physical Demand	28	0.7857143	0.9946949
Temporal Demand	28	2.8928571	1.2572541
Performance	28	4.0000000	0.9813068
Effort	28	2.4642857	1.2013000
Frustration	28	2.1071429	1.5236235

Table 17: Means and std deviations on workload factors rates (0-20) for failure reporting experiments.

Criterion	count	mean	sd
Mental Demand	28	10.500000	4.826624
Physical Demand	28	4.678571	4.753584
Temporal Demand	28	7.428571	4.646214
Performance	28	12.214286	4.532656
Effort	28	11.214286	4.466809
Frustration	28	9.357143	5.736267

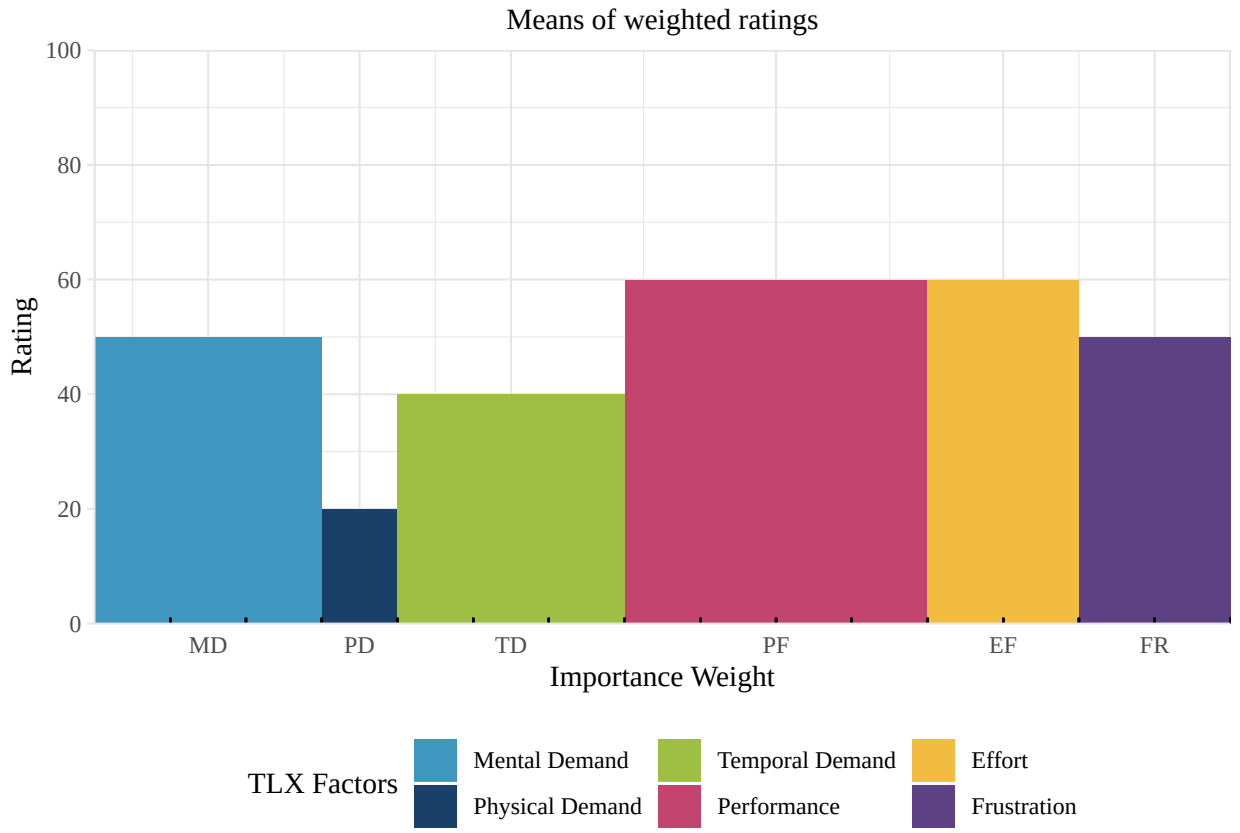


Figure 9: NASA-TLX weighted rates plot on workload factors in diagnosis reporting experiments.

4.5. Usability results

Usability is defined as a qualitative measure regarding the degree to which alternative solutions (ARR, ARN, TBR, TBN) enhance completion of diagnosis reporting tasks. Based on Nielsen's usability criteria **[ref]**, Table 5 described surveyed usability criterions and solutions' aspects against to which assess those. In order to confirm experimental results, testers were hypothesised to perceive usability of AR reporting solutions (ARR, ARN) at least as good as non-AR solutions (TBR, TBN) with small variances between recommender and non-recommender ones.

Figure 10 and Table 18 summarise testers' responses for each usability criterion per reporting solution. Average criterions responses range from 3.24 to 4.26 in a Likert Scale 1-5 with higher variabilities for ARR and TBN solutions. Ease-To-Learn was the lowest scored criterion with averages between 3.24 (ARR) and 3.64 (ARN). Ease-To-Use was the criterion with lowest variability ranging from 3.81 (ARR) and 4.21 (TBR) with

TBN showing the highest variability. Effectiveness and Satisfaction responses were higher for AR solutions ranging respectively between 3.47 (TBN) and 4.14 (ARN), and 3.57 (TBR) and 3.93 (ARN). TBN and TBR solutions were better perceived regarding Ease-To-Use, while ARR and ARN were better at Effectiveness and Satisfaction. Further analyses on these criterions can be done studying different solutions' aspects they are affected by.

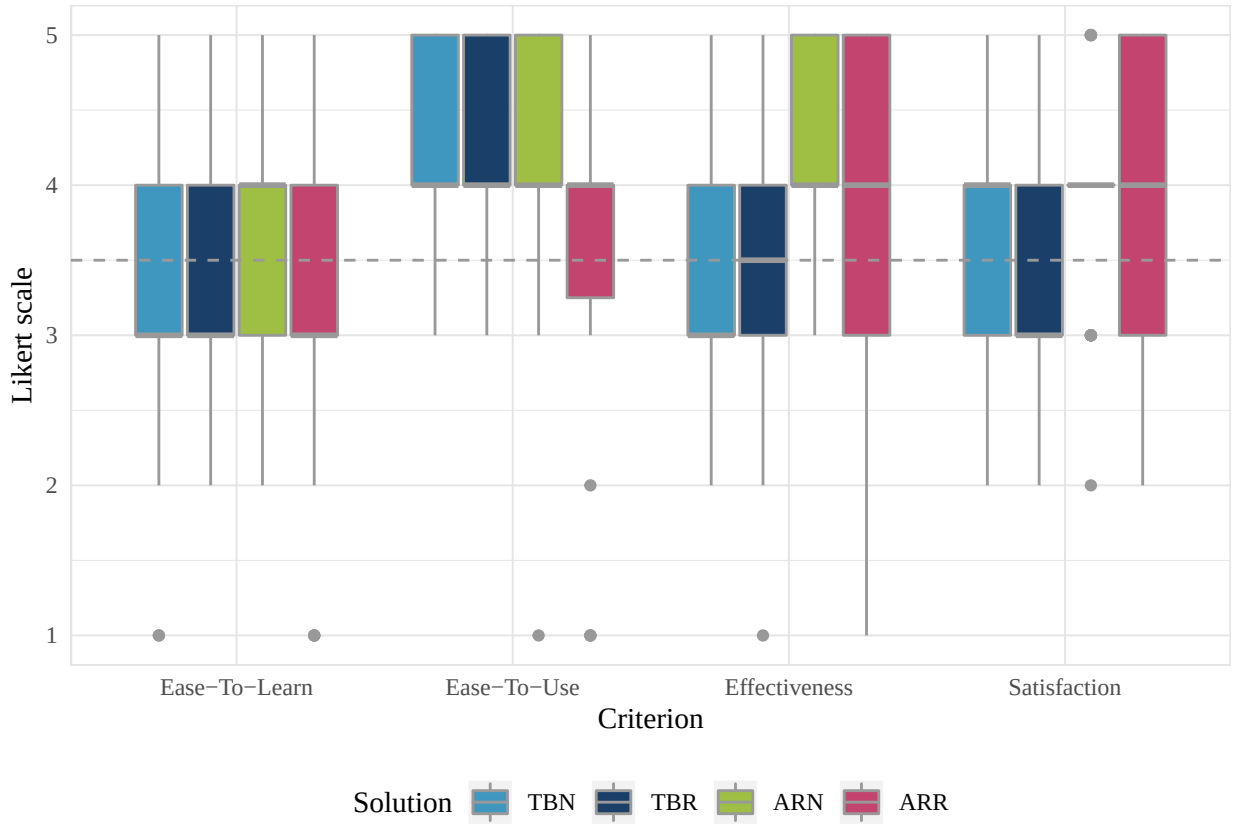


Figure 10: Box and whiskers plot on testers responses for usability criterions per solution.

Figure 11 presents average testers' responses regarding Ease-To-Learn aspects of alternative solutions (TBN, TBR, ARN, ARR). These results suggest that AR solutions (ARR, ARN) have higher learning curves than tablet-based solutions (TBR, TBN) according to the differences between ease at start and at end. Also, AR solutions seem more intuitive than tablet-based ones.

Table 18: Means and std deviations on testers responses for usability criterions per reporting solution.

Criterion	Solution	count	mean	sd
Ease-To-Learn	TBN	42	3.333333	1.0280616
Ease-To-Learn	TBR	42	3.404762	1.0373400
Ease-To-Learn	ARN	42	3.642857	0.9833102
Ease-To-Learn	ARR	42	3.238095	1.0777013
Ease-To-Use	TBN	42	4.214286	0.7168942
Ease-To-Use	TBR	42	4.261905	0.7344991
Ease-To-Use	ARN	42	4.071429	0.8379085
Ease-To-Use	ARR	42	3.809524	1.0646904
Effectiveness	TBN	70	3.471429	0.8634516
Effectiveness	TBR	70	3.571429	0.9413439
Effectiveness	ARN	70	4.142857	0.7668062
Effectiveness	ARR	70	3.585714	1.3020926
Satisfaction	TBN	42	3.880952	0.8323455
Satisfaction	TBR	42	3.571429	0.8006966
Satisfaction	ARN	42	3.928571	0.6005224
Satisfaction	ARR	42	3.857143	1.0722993

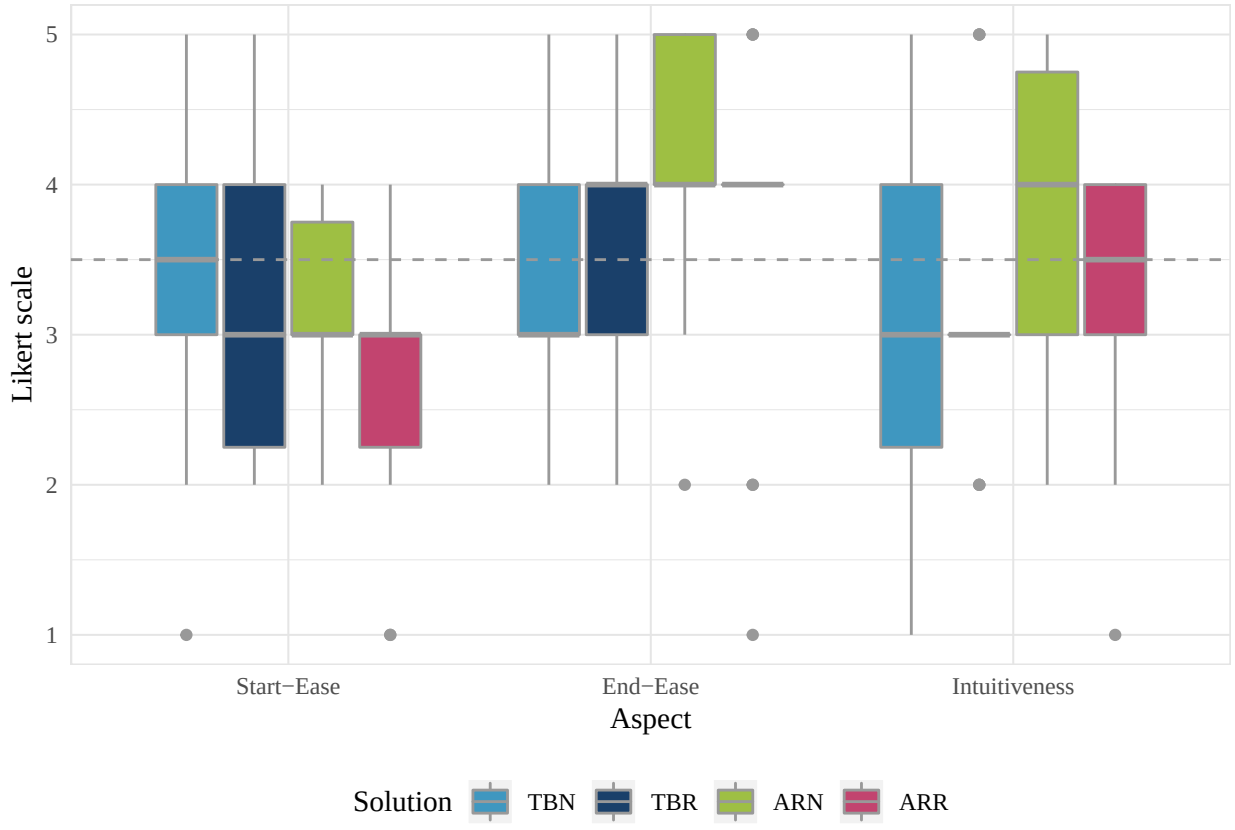


Figure 11: Box and whiskers plot on testers responses for Ease-To-Learn per solution.

Figure 12 presents average testers' responses regarding Ease-To-Use aspects of alternative solutions (TBN, TBR, ARN, ARR). Ease-To-Use aspects refer to solutions' user interface items.Gestures and text of AR

solutions were perceived lower than buttons and text from tablet-based tools. Instead, testers perceived AR dictation capabilities better for data input than normal tablet keyboard.

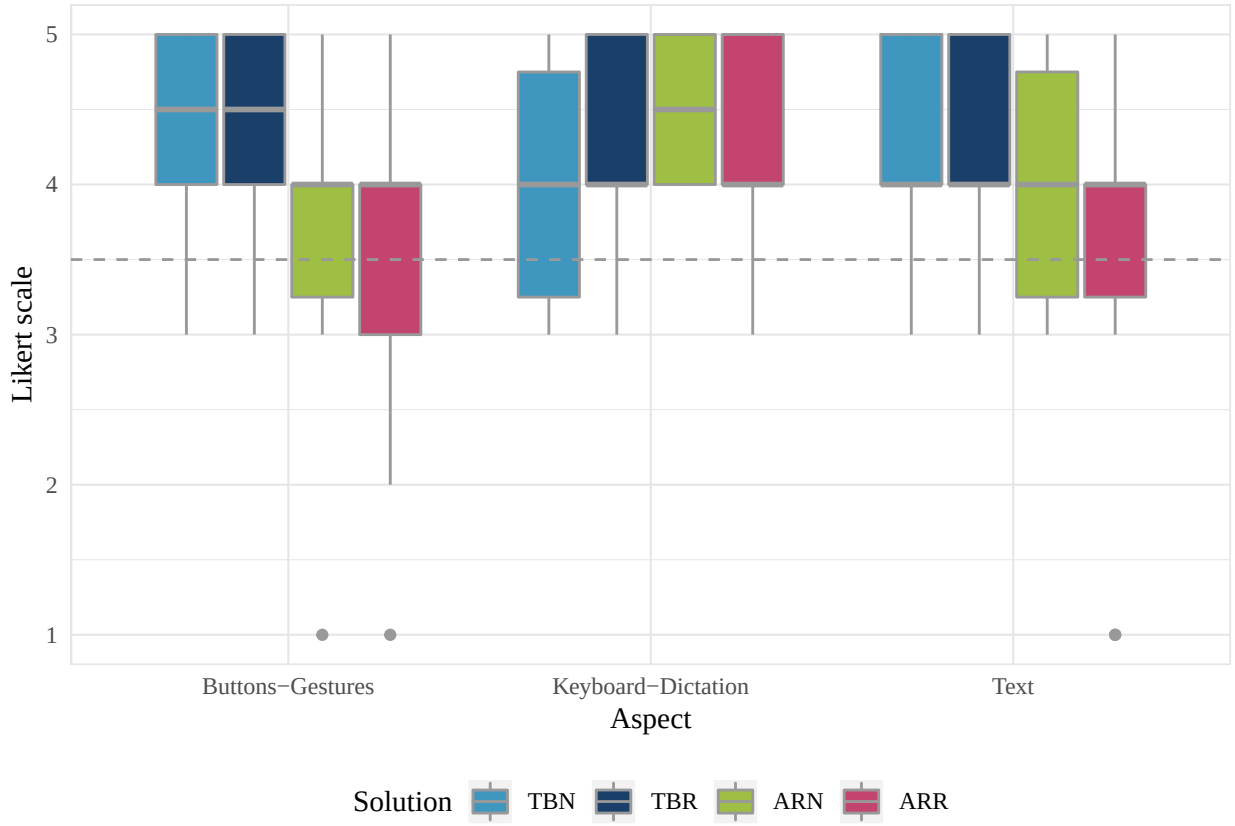


Figure 12: Box and whiskers plot on testers responses for Ease-To-Use per solution.

Figure 13 presents average testers' responses regarding Effectiveness aspects of alternative solutions (TBN, TBR, ARN, ARR). AR solutions scored higher in certain aspects such as ease to understand, efficiency and confidence increase and content suitability. Besides, AR and non-AR solutions had similar testers responses for error reduction and report accuracy. ARR is the solution with higher variabilities.

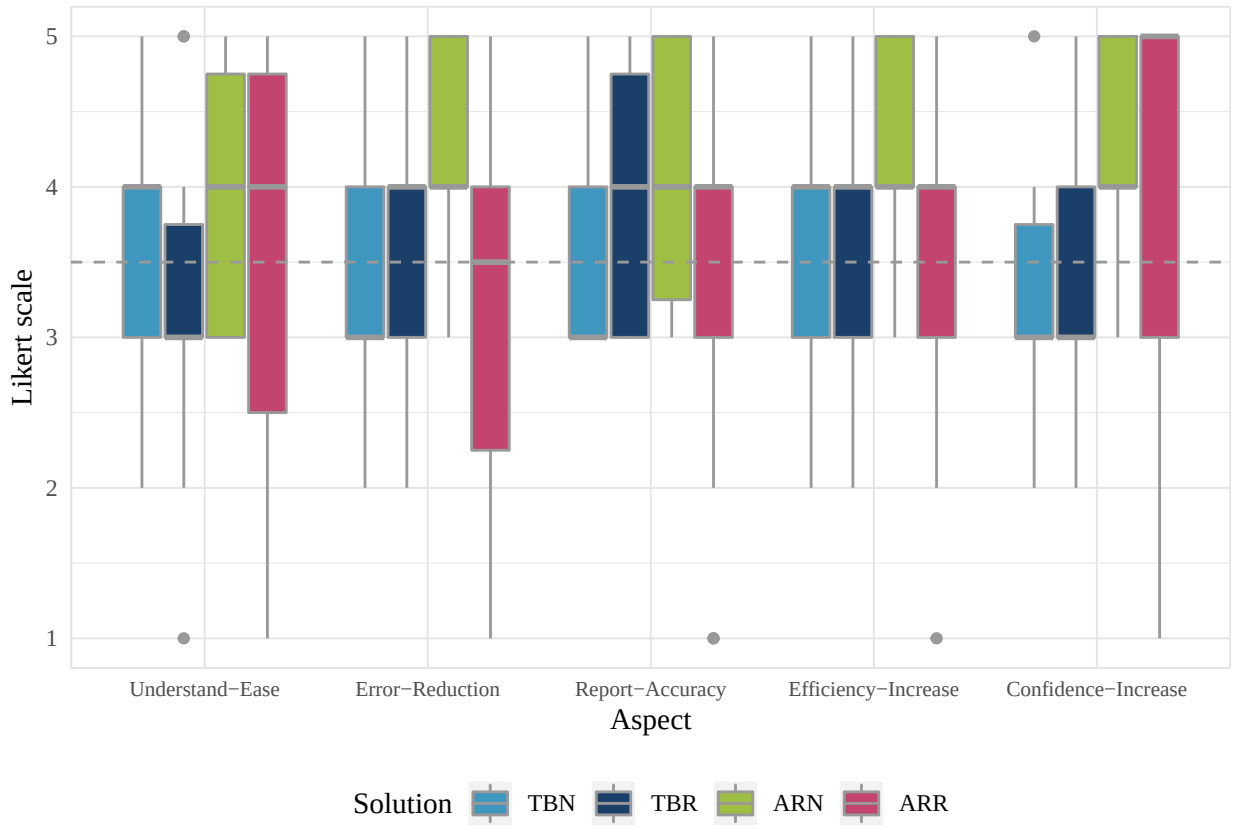


Figure 13: Box and whiskers plot on testers responses for Effectiveness per solution.

Figure 14 presents average testers' responses regarding Satisfaction aspects of alternative solutions (TBN, TBR, ARN, ARR). Testers perceived design of different solutions very similarly. Instead, feeling and overall satisfaction of AR solutions (ARR, ARN) was better perceived by testers than tablet-based ones.

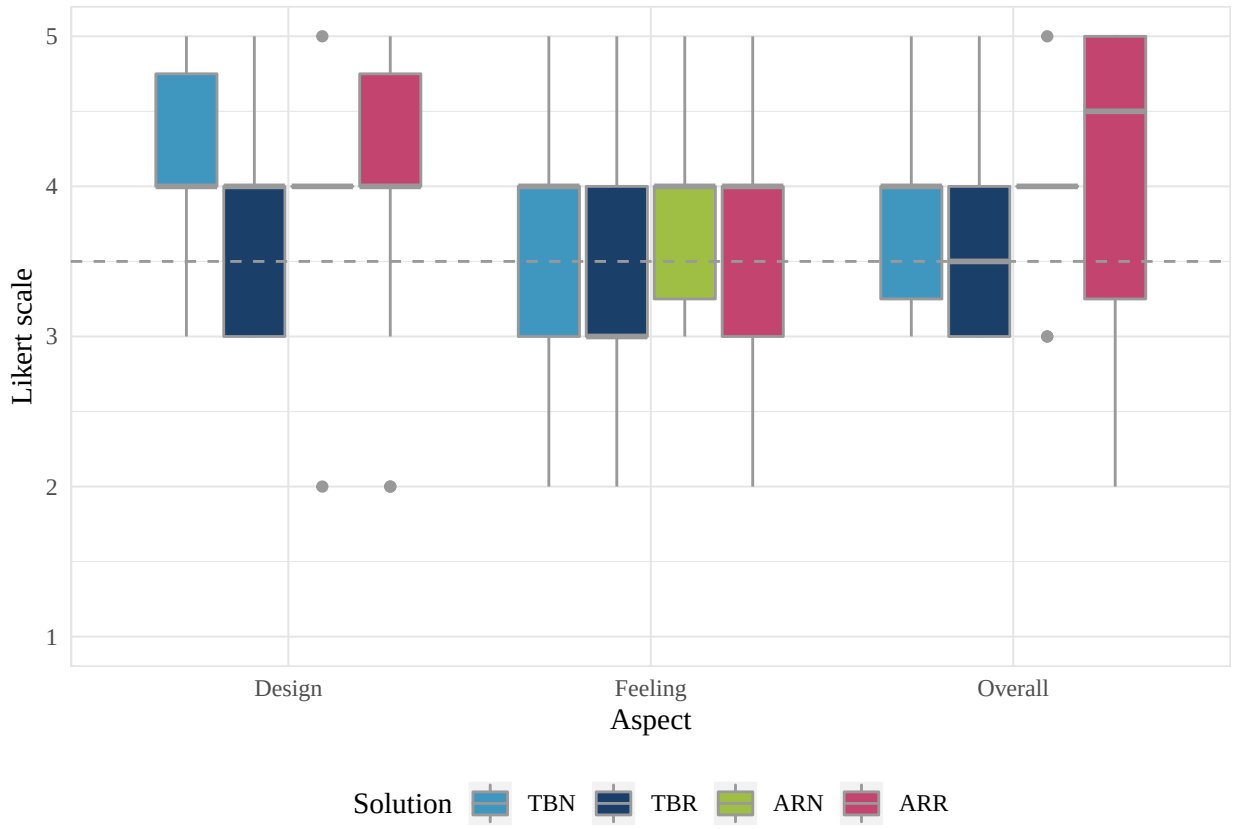


Figure 14: Box and whiskers plot on testers responses for Satisfaction per solution.

Overall, usability surveys indicate that testers perceived AR solutions as more effective and satisfactory, while non-AR solutions were perceived as easier to learn and understand. According to these results, AR solutions showed higher learning curves than non-AR tools, although advanced data input methods (e.g. dictation) were also well considered by testers. Besides, AR solutions were percieved as more accurate, enhanced and assuring for diagnosis reporting operations. These results indicate validity of this research's hypothesis regarding improved usability of AR reporting solutions, but do not suggest significant differences among recommender and non-recommender solutions.

5. Conclusions

5.1. Discussion

Previous analytical results aimed to evaluate this research's hypothesis (**Section 2**) for demonstrating the validity of this research's contributions (**Section 1**).

The first validation hypothesis stated that "recommendations accuracy improves with the use of AR-based hybrid recommendations compared to conventional ontology-based methods". Accuracy effect study (**Section 4.1**) analysed the relation between recommendations proposed and chosen by testers and recommender methods in diagnosis reporting experiments. Its results showed that the proposed method's accuracy (ARR) was 2.2 times higher than conventional recommender's (TBR) on average for both experimental failures. T-tests results indicated a statistically significant difference on accuracy results per method. Therefore, this hypothesis can be considered valid within the context of the experiments conducted. Due to measures' nature, identified accuracy improvements can be the result of: (1) more precise recommendations and (2)

more correct testers' selections caused by contextualised visualisations. Future studies can investigate the independent effects of each cause in real-life experiments to quantiatively measure their independent impacts.

Following validation hypotheses assumed that "errors are a measure of reporting effectiveness and time is a measure of reporting efficiency". Pearson's correlation test's results (**Section 4.2**) indicated that the correlation between these response variables could not be considered statistically significant. Moreover, Cohen's interpretation of correlation test's results suggested that even if correlation was significant, it was small. Hence, it can be said that within the context of this research's experiments the assumption above can be considered valid.

The second validation hypothesis enounced that "errors reduce with AR reporting solutions compared to non-AR ones". And the third one stated that "errors reduce with recommender reporting methods comapred to non-recommender ones". Errors study (**Section 4.2**) analysed the effect on errors rate per reporting task of diverse reporting tools (ARR, ARN, TBR, TBN) in failure diagnosis experiments. Errors rates for AR-based tools (10%) reduced 50% compared tablet-based solutions' errors rates (20%) in both experimental failures. Besides, two-way ANOVA test results indicated statistical significance of these differences within the experiments' context. Therefore, the second hypothesis can be considered valid but not the third one. Due to the nature of experiments, errors were measured per reporting task. So, a possible explanation for these conclusions is that AR methods allow to contextualise complex data input tasks but recommender methods reduce their total number of tasks. Future works can extend recommendation facets to different datasets for studying the effect of recommendations in singular data-input tasks.

The fourth validation hypothesis stated that "time decreases with the use of AR reporting solutions compared to non-AR ones". And the fifth one enounced that "time decreases with the use of recommender reporting tools compared to non-recommender ones". Time study (**Section 4.3**) evaluated the effect on time of AR and recommender reporting and their counterparts in failure diagnosis experiments. In electric failure experiments, the proposed AR-recommender reporting solution (ARR) was 20% faster than its non-recommender counterpart (ARN) and 23% faster than non-AR tools (TBR, TBN). In electronic failure experiments, ARR was 22% faster than ARN, 26% faster than TBR and 37% faster than TBN. Besides, two-way ANOVA test results indicated a significant effect of solution and failure factors on experimental time results. Post hoc comparisons from Tukey's HSD tests confirmed that resultant differences were mostly driven by the variances between the proposed reporting tool (ARR) and different alternatives. Overall, these results suggest the validity of both time-related hypothesis together as the proposed AR-recommender solution was found faster than its counterpart in diagnosis reporting experiments. They also suggest the need to correlate AR content formats and recommendations to improve efficiency of knowledge capture applications. Future studies can investigate this correlation more in-depth to quantitatively measure their independent effects. They can also further corroborate this research's results applying the proposed methods to other AR-maintenance knowledge capture applications.

The sixth validation hypothesis stated that "testers' should percieved experimental workload as mentally and performance demanding". Thus, aiming to corroborate previous hypothesis regarding improvements on reporting effectiveness and efficiency. Surveys results (**Section 4.4**) evaluate NASA-TLX criterions regarding tasks workload requisites. These suggested that testers perceived reporting tasks mostly as performance and effort demanding, with temporal demand and frustration factors following closely. Nevertheless, post-experimental surveys for scoring factor rates were done after testers completed both experiments with alternative AR and non-AR solutions. Also, testers were novices with little maintenance experience. Future works can investigate the difference in percieved workload with different solutions and with real-life maintainers to further clarify these tasks' requisites.

The final validation hypothesis enounced that "perceived usability improves for reporting tools implementing AR and recommender methods". Thus, aiming to corroborate previous hypotheses regarding improvements on reporting effectiveness and efficiency. Surveys results (**Section 4.5**) evaluated usability criterions according to different reporting tools' aspects. These indicated that testers' perceived AR solutions as more effective and satisfactory, while non-AR solutions were perceived as easier to learn and understand.

5.2. Conclusions

This paper proposed (1) a method to automatically create and dynamically allocate AR content for maintenance knowledge capture applications and (2) a method to provide AR content with context-aware, ontology-based recommendations to simplify knowledge capture procedures. Their aim was to prove that automatic recommendable authoring can improve efficiency and effectiveness of AR knowledge capture applications. They were implemented in a cloud-based AR application prototype for validation with reporting effectiveness and efficiency experiments and usability surveys on reporting diagnosis operations. Experimental results indicated that the proposed AR-recommender reporting method reduces reporting errors (50%) and time (20%) compared to alternative non-AR and non-recommender solutions. These results also displayed that recommendations' accuracy doubles for the proposed AR-based hybrid techniques compared to conventional ontology-based methods. Besides, surveys results suggested that testers' perceived the proposed reporting solution as more effective and satisfactory than its non-AR and non-recommender counterparts. Thus, proving that the proposed methods can improve effectiveness and efficiency of diagnosis reporting applications.

The proposed methods for automatic recommendable auhtoring contribute to fill an important research gap towards the integration of human operations in digital maintenance. Maintenance reporting operations are performance and mentally demanding and so, prone to errors in efficiency-challenging conditions. These often result on reports with decreased accuracy and unstructured knowledge difficult to re-use. Through contextualisation and standardisation of data input tasks, these methods can enhance the digitalisation of maintenance reporting operations. Thus, facilitating the integration of human knowledge in digital maintenance.

5.3. Future works

Future works will explore further applications and enhancements of the proposed methods for pursuing human knowledge integration in digital maintenance. The following list extends the future works described within this paper's discussions:

- Dynamic content allocation:
  - Investigate factors that can cause occlusion in AR maintenance applications and improve proposed allocation and scaling mechanisms to reduce it.
  - Study dynamics of AR knowledge capture applications and improve proposed allocation mechanisms to enhance content navigation.
- Content formats:
  - Study dynamics of maintenance knowledge capture applications and improve content formats adaptability to enhance simultaneity of knowledge transfer and capture.
  - Develop advanced methods to determine input data correcteness for further reducing reporting errors and improving effectiveness.
  - Develop advanced content formats to report heterogenous and unstructured data types (e.g. audio or images) for further integration of human knowledge in digital maintenance.
  - Develop adaptive content formats according to user and environmental conditiongs (e.g. performance or light) for further decreasing reporting time and improving efficiency.
- Recommendation facets:
  - Extend proposed recommendation framework to different techniques (e.g. collaborative filtering) and implement automatic data collection methods (e.g. content-tracing or eye-tracking) for improving recommendations accuracy.
- Applications and experiments:
  - Experiment with the proposed methods and real-life maintainers in real-life conditions to study the correlation between AR content and recommendations in maintenance reporting operations and study their independent effects on reporting workload, errors and time.
  - Develop new content formats and recommendation facets for different maintenance reporting operations (e.g. service logs) to extend integration of human knowledge in digital maintenance.

Augmented Reality technologies are information visualisation tools that can smooth knowledge transfer between humans and digital systems. These future works aim to find the necessary research towards the integration of human knowledge in digital maintenance. Thus, envisioning a future where maintenance digital systems can re-use human knowledge to its full extent.

Appendix A. Future works draft

- Dynamic content allocation:
- automatically adapt elements positions and scales to eliminate occlusion
- improve content allocation mechanisms to improve interface navigation and enable consulting and reporting applications
- Content reporting:
- limitations on property assertion need development of adaptive content formats to instantiate properties more than once
- advance data input methods and correctness procedures to further reduce reporting errors
- develop new formats to report more advanced datatypes like images and audio and provide additional results
- improve formats contextualisation to user and environment (e.g. user performance, light conditions)
- Recommendation techniques:
- extend facets to enable user collaborative filtering recommendations including automatic data capture methods like eye-tracking, content-tracing, etc.
- Further experiments:
- experiment the proposed solution in real-life conditions to measure independent impact of AR content formats and recommendations in accuracy
- experiment the proposed solution with real-life maintainers to measure independent impact of AR content formats and recommendations in workload, errors and time
- extend the proposed facets for other maintenance reporting applications and corroborate this research's results

Appendix B. Ideas for future works

- Discuss that numbers chosen have been so according to authors tests. Nevertheless, future works should investigate these in more depth to ensure that no occlusion occurs with any possible combination of elements displayed. These may include algorithms to solve this geometrical problem dynamically.
- Include in discussion the facts that only one property assertion is allowed per instantiated individual. This is due to the use of RDFS language for ontology declaration in neo4j. For the case of study, it has not been a problem because no property declared was meant to be asserted more than once in an individual. Future works should aim to solve this issue and include more adaptive content formats to instantiate a property more than one.
- Future works that develop new content formats for knowledge capture are required to do the same. Value types may be necessary when creating more advance content types like images or audio. So, PMAU algorithm can infer what type of file format is required.
- Future works will study alternative recommendation approaches (e.g. content-based diversity, collaborative-filtering) and implement advanced techniques for automatic data collection (content-tracing, eye-tracking) to enhance them.
- Future works: demonstrate validity of the proposed technique to further adapt augmented content to the context (e.g. users and environment).
- Future studies can investigate theindependent effects of each cause in real-life experiments to quantiatively measure their independent impacts
- Future works can extend recommendation facets to different datasets forstudying the effect of recommendations in singular data-input tasks.
- Future studies can investigate this correlation more in-depth to quantitatively measure their independent effects. They can also further corroborate this research's results applying the proposed methods to other AR-maintenance knowledge capture applications
- Future works can investigate the difference in percieved workload with different solutions and with real-life maintainers to further clarify these tasks' requisites.

- IMPROVE RECOMMENDATION ALGORITHMS: INCLUDE FORMATS FOR USER-FILTERING, USER PERFORMANCE, ETC.
- IMPROVE DATA INPUT CONTENT FORMATS: INCLUDE MORE ADVANCED RULES FOR DATA CHECKING, ETC.
- IMPROVE VISIBILITY OF REPORTED INDIVIDUALS: INCLUDE PANEL WITH SUMMARY
- IMPROVE NAVIGABILITY AMONG INDIVIDUALS: IMPROVE NAVIGATION INTERFACE
- EXTEND RECOMMENDATION FACETS TO OTHER REPORTING CLASSES (E.G. STATES, ETC.)
- EXTEND METHODS TO OTHER REPORTING APPLICATIONS (E.G. SERVICE LOGS, ETC.)
- INCLUDE ADDITIONAL ALLOCATION METHODS FOR ALLOCATABLE CONTENT (E.G. STICKY NOTES, ETC.): STUDY MERGE OF TRANSFER AND CAPTURE APPLICATIONS (DIAGNOSIS AND REPORTING)