Prediction of Human Factors in Aviation Incident Reports using Machine Learning and Natural Language Processing

Tomás Moitinho de Almeida Andrade Madeira
tomas.madeira@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

January 2021

Abstract

In the aviation sector, human factors are the primary cause of safety incidents. Intelligent prediction systems, capable of evaluating human state and managing risk, have been developed over the years to identify and prevent human factors. However, the lack of large useful labelled data has often been a drawback to the development of these systems. This paper proposes and implements a predictive model that can identify and classify human factor categories from aviation incident reports. The study data, provided by the Aviation Safety Network (ASN), comprises 1674 incident reports between 2000 and 2020. These reports consist of aircraft details, planned route, probable cause, and other flight information. A novel human factor classificatory framework is proposed and a diversified labelling set is developed, based on the acquired data. For feature extraction, a text pre-processing and Natural Language Processing (NLP) pipeline is developed. For data modelling, semi-supervised Label Spreading (LS) and supervised Support Vector Machine (SVM) techniques are considered. Random search and Bayesian optimization methods are applied for hyper-parameter analysis and improvement of model performance, measured by the Micro F1 score. The best predictive models achieved a Micro F1 score of 0.900, 0.779 and 0.875, for each level of the proposed framework, respectively. The proposed solution indicates that favourable predicting performances can be achieved for the classification of human factors based on text data. Notwithstanding, a larger data set would be recommended in future research.

Keywords: Machine Learning, Natural Language Processing, Classification, Human Factors, Aviation Safety

1. Introduction

Over the past decades, human factors have been the main latent cause of aviation safety breaches. Studies such as [1, 2, 3] have found pressure, fatigue, miscommunication, and lack of technical knowledge on crucial personnel - such as maintenance workers, air crew, and air traffic controllers - to be some of the main probable causes for aviation mishaps. According to the International Civil Aviation Organization (ICAO), in 2018 alone, there were 98 aircraft accidents for scheduled commercial air transport operations, of which 11 were fatal accidents, resulting in 514 passenger fatalities [4]. These figures reflect the current relevance of human factor predictive safety models that can detect and prevent high-risk situations for the aviation sector.

Recently, it is notable an increase in a common effort within the research community to develop data-based Human Reliability Assessment (HRA) processes that can produce accessible predictive indicators, while leveraging already acquired data. However, some of these prominent processes [5, 6], especially those which rely on the contents of text reports, often require manual categorization of human factor categories, an expensive and error-prone task.

The aim of this research is to contribute to a better knowledge about how to enhance aviation safety, by developing a comprehensive methodology based on data mining and machine learning techniques, to identify and classify the main human factors causal of aviation incidents, based on descriptive text data.

The general problem of inferring taxonomic information from text data is not novel and has been extensively explored in other fields of research, such as healthcare and journalism. Some examples of successful applications have been the prediction of patient illness based on medical notes [7, 8] and automated fake news detection from internet pages [9]. Surprisingly, to our knowledge, only a few studies have tried to infer information from aviation safety reports using NLP [10].

This paper is organized as follows. Section 2 presents a carefully conducted initial data analysis and pre-processing of the corpora, and introduces a novel HFACS-ML framework to facilitate human factor classification on machine learning applications. Moreover, a diversified labelled set is also developed. After that, Section 3 describes how embedding techniques can be used to associate the semantic meaning between long pieces of text by comparing, in a local setting, human factor categories of differently distanced documents. All the work developed in Sections 2 and 3 is availed in Section 4, where we associate the labelled samples and document vectors with classification algorithms in order to infer the category of unknown documents. In a preliminary analysis, using a D2V and LS combination, we gain insight into some of the limitations that may corrupt our models, and iterate on this information to improve over the different levels. Then, in section 5 conclusions are addressed, as well as some recommendations for future work.

2. Tailored Data Analysis

In order to acquire descriptive texts containing the most recent threats to aviation safety, for this study, we gathered the last two decades (2000 to 2020) of ”Probable Cause” reports from the publicly available ASN database, amounting to a total 1674 documents. Additional information on the database and report structures can be found in [11].

2.1. Human Factor Classification Framework

After a comprehensive examination of the database, it resulted clear that the content present in the text reports could not be exactly correlated to the standard Human Factor Analysis and Classification System (HFACS), shown in Figure 1. Some reasons for the inapplicability of this framework were: lack of Organizational Influence information; insufficient subcategorical detail; and confliction between core vocabularies within the same category.

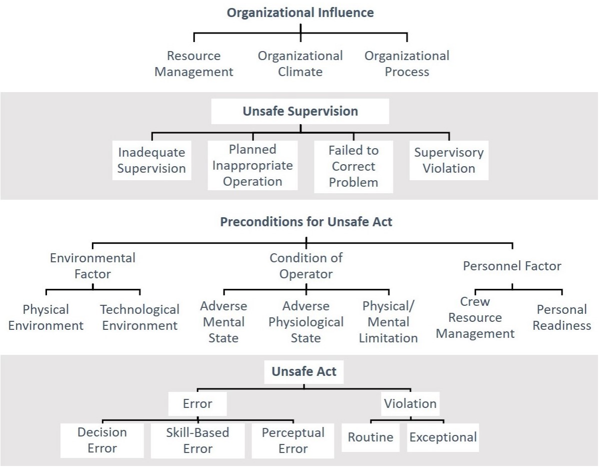

Figure 1: HFACS framework (adapted from [12]).

For this reason, a variation from this framework, adapted for machine learning (ML) research, the HFACS-ML, was proposed (Figure 2). This new framework was designed to correct the previously mentioned challenges, as well as to facilitate the association between the various distinguished contexts found in the ”Probable Cause” reports to independent human factor categories. One main change worth pointing out was the division of the ”Physical Environment” category into two distinct categories, ”Physical Environment 1” and ”Physical Environment 2”, appurtenant to weather and animal preconditions, respectively. Another change worth noticing was the introduction of outlier categories, ”Not Available” (n/a) and ”Undetermined” (und), for the cases where no human factor would be mentioned in the text or the cause of the incident was explicitly undetermined. Additional descriptions of the remaining categories can be found in [12].

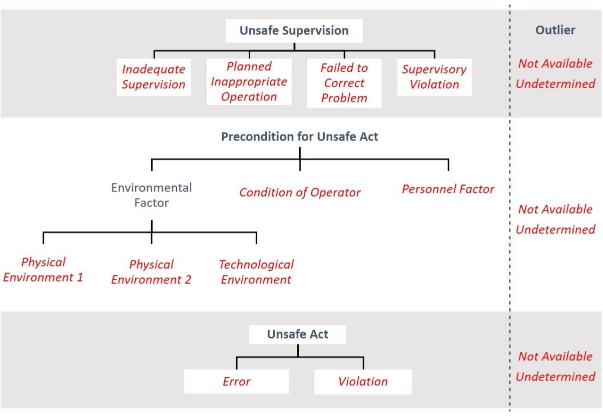

Figure 2: Proposed HFACS-ML framework.

Note that, similarly to the original HFACS, in the proposed HFACS-ML each document may have a minimum of zero labels and a maximum of three labels, with at most one label per level.

2.2. Construction of a Labelled Data Set

To enable the development and testing of predictive classification models, we constructed a labelled set using two simple and efficient approaches: Data-driven automated labelling and manual labelling.

In the first approach, we used keywords available in the database, already attributed to some of the documents, and searched for possible associations with HFACS-ML categories. For this task, a consistency criterion was defined: in the observation of 15 random documents with a certain keyword, if at least 12 belonged to the same HFACS-ML category, for a certain level of the framework, then consistency was satisfied. In these cases, all documents which possessed that keyword would be equipped with the same human factor label for that particular level, and the observed irregular samples would be manually corrected. If the consistency criterion was not satisfied for a certain keyword, then no label would be attributed to any of the respective reports. Table 1 shows all keyword associations that were found to satisfy consistency, and therefore contributed to the data-driven automated labelling.

Table 1: Automated labelling table

	HFACS-ML Level		
Keyword	**Unsafe Supervision**	**Precondition**	**Unsafe Act**
Weather - (all)	-	Physical Env. 1	-
ATC & navigation - VFR flight in IMC	-	Physical Env. 1	-
ATC & navigation - Language/communication	-	Personnel Factor	-
Collision - Object - Bird	-	Physical Env. 2	-
Collision - Object - Person, animal	-	Physical Env. 2	-
Airplane - Engines - Fuel exhaustion	Supervisory Violation	Personnel Factor	-
Airplane - Engines - Fuel starvation	Supervisory Violation	Personnel Factor	-
Flightcrew - Alcohol, drug usage	-	Condition of Operator	-
Flightcrew - Incapacitation	-	Condition of Operator	-
Flightcrew - Disorientation, sit. awareness	-	Condition of Operator	Error
Flightcrew - Insufficient rest / fatigue	-	Personnel Factor	-
Flightcrew - Non adherence to procedures	-	-	Violation
Cargo - Overloaded	-	-	Violation
Flightcrew - Un(der)qualified	Supervisory Violation	Personnel Factor	-
Security - Suicide	-	Condition of Operator	Violation

Although a considerable amount of labels was attained through this labelling method, the distribution of the resulting set revealed very imbalanced. For this reason, in order to add variety to the labelled set a second approach, manual labelling, was also conducted. Throughout the course of this study, more than 60 documents were individually analyzed and classified onto their respective HFACS-ML categories. The result of both labelling processes lead to a total classification of 107 Unsafe Supervision labels, 370 Precondition for Unsafe Act labels and 119 Unsafe Act Labels. The complete label distribution is summarized in Table 2.

Table 2: Total label distribution.

HFACS-ML Level	HFACS-ML Category	Label count
Unsafe Supervision	Inadequate Supervision	20
	Planned Inap. Oper.	6
	Failed Known Prob.	3
	Supervisory Violation	52
	n/a	22
	und	4
Precondition for Unsafe Act	Physical Env. 1	169
	Physical Env. 2	46
	Technological Env.	10
	Condition of Operator	55
	Personnel Factor	78
	n/a	8
	und	4
Unsafe Act	Error	54
	Violation	47
	n/a	14
	und	4

2.3. Pre-Processing

In data mining, the presence of irrelevant information, often found in raw text data, is known to substantially condition the performance of predictive models. Since, to our knowledge, no studies have tried to explore which pre-processing tools result most efficient for aviation incident report analysis, we took inspiration from studies applied to other settings, such as [13, 14], to implement a tailored pipeline. The resulting process can be summarized in three stages: Data cleaning, Normalization and Tokenization.

In the first stage, all duplicate instances were removed and all incidents which originated from terrorist assaults were excluded. The reason behind the latter was based on the principle that personnel performance under malicious external threats should not be representative of their professional behaviour under conventional circumstances.

In the second stage, all non-English documents were translated into English, all letters were lowercased, as suggested in [13], and punctuation was removed.

In the third stage of pre-processing, for each document, the text was parsed (or tokenized), converting each word into a single entity (or token). For this step, we chose to apply alphabetic parsing and stripped all digits from the data set. Although significant information may at times be derived from these characters, we found them not to provide any additional value regarding human factors, as the main relevant semantic meaning from our database was often found in word descriptions and core vocabularies. The same justification applies to punctuation removal.

After parsing, we considered the removal of stop-words. For this purpose, two lists of unwanted words were introduced. The first list, extracted from the publicly available [15], consisted of standard stop-words commonly used for the treatment of natural English data. The second list, was tailored to our data set and designed to handle introductory information, which could appear in different parts of the text. This list consisted of the following words: ‘summary’, ‘probable’, ‘cause’, ‘accident’, ‘contribute’, ‘factor’, ‘find’, ‘conclusion’, ‘translate’, ‘spanish’, ‘italian’, ‘french’, ‘german’.

In the final step of this stage, words underwent lemmatization, a morphological process which leverages dictionary information to reduce words to their base form. This process is especially useful for feature extraction as it simplifies the vocabulary and facilitates semantic word association. Following this process, extremely rare words appearing 5 or less times throughout the corpora were also ignored, as these would prove too rare to form meaningful patterns.

Together, all the above pre-processing steps provided a significant contribution to improving data quality and reducing computational costs, by homogenizing the text and reducing noisy or unwanted information. The result of this process is availed in the next section.

3. Feature Extraction with NLP

In order to enable computers to read, decipher and understand the semantic meaning of language data in a manner that is valuable, mathematical models are used to convert text segments into numerical vector projections. This process is referred to as feature extraction. A series of NLP models, specifically designed to process natural language data, have been considered to efficiently derive document projections from our "Probable Cause" reports.

3.1. TF-IDF

In the Term Frequency–Inverse Document Frequency (TF-IDF) algorithm, each feature in a document vector is associated to a single word from the vocabulary, and increases proportionally to the frequency of that word in the same document. However, the value of this feature is also offset by the number of documents in which that word appears. The latter concept helps to adjust for the fact that some words appear more frequently in general, and should therefore be considered with less importance, while others more domain specific should be compensated with a greater weight [16].

Formally, let $V = \{w_1, w_2, ..., w_V\}$ be the set of distinct words in the vocabulary, each feature $q_i(w)$ in a TF-IDF document vector d_i represents the weight word w possesses for that document. Additionally, let $f_i(w)$ be the frequency of the same word, in the same document, and $f_N(w)$ be the total number of documents in which that word appears. The formal weight computation is shown in equation 1.

$$q_i(w) = f_i(w). \log \frac{N}{f_N(w)} \tag{1}$$

Although the simplicity of this algorithm, in this overview we also present some of its biggest limitations: computational complexity increases with the size of the vocabulary; word relations are not captured; and it has trouble handling out of vocabulary words, for the classification of new documents [17]. For these reasons, the next subsections describe other approaches used in this research that seek to solve some of the above mentioned obstacles.

3.2. Word2Vec

Also known as W2V, this feature extraction algorithm, introduced in [18], uses shallow Neural Networks (NN) to efficiently derive word embeddings (or vectors) of custom size P. It can do so through two different architectures: the Skip-Gram (SG) and the Continuous Bag of Words (CBoW).

In the first architecture, a shallow Neural Network is trained on the task of predicting the surrounding context words $w_{O,i}$ of a single target word w_I, given a context window of size C. The objective function is shown in equation 2.

$$log\, p(w_{O,1}, ..., w_{O,C}|w_I) \tag{2}$$

In the second architecture, a shallow Neural Network is trained on the task of predicting a single target word w_O, given a set of context words $w_{I,i}$ and a context window of size C. Equation 3 shows the respective objective function.

$$log\, p(w_O|w_{I,1}, ..., w_{I,C}) \tag{3}$$

After the training process, word embeddings can be extracted from the weights of the hidden layer and transformed into document embeddings. Let v_w be the vector projection of word w, d_i the vector projection of an arbitrary document i, and T_i be the set of ordered words found in that same document. Equation 4 shows the mathematical equivalent of this transformation.

$$d_i = \frac{1}{|T_i|} \sum_{w \in T_i} v_w \tag{4}$$

However, another more recent method enables the computation of document embeddings directly from NN training. This method is described in the following subsection.

3.3. Doc2Vec

First proposed in [19], this feature extraction algorithm introduces 'paragraph vectors' that act as memory devices which retain the topic of paragraphs. In our case, we use these vectors to directly portray the information from the text documents in a vector of custom size P. The underlying intuition of Doc2Vec (D2V) is that document representations should be good enough to predict the words or the context of that document.

This algorithm's two principal architectures, the Distributed Memory (DM) and the Distributed Bag of Words (DBoW), hold a high affinity to W2V's CBoW and SG architectures, respectively. The main difference is that the document vectors from this new algorithm are directly embedded in the NN training and prediction.

3.4. Preliminary Analysis

In this subsection we analyse, at a local level, how human factor categories may be inferred from the document projections generated by the previously described architectures. For this purpose, we used the widely known cosine similarity measure (equation 5) to identify the 5 closest and 3 furthest documents from a randomly picked report, with identification (iD) 361, and analysed them as to their human factors.

$$S(d_1, d_2) = cos(d_1, d_2) = \frac{\vec{d_1}.\vec{d_2}}{||\vec{d_1}||.||\vec{d_2}||} \tag{5}$$

Tables 3 and 4 illustrate the best results from this test, provided by the D2V DM model.

The set of tests conducted in this section suggest that documents with similar human factors may tend to be placed in close cosine distances, while documents with distinct human factors might tend to be placed further apart. This observation justifies the set of models, described in the next section, designed to classifying human factors of unknown documents based on their position in the vector space.

Table 3: Most similar documents, from the D2V DM model.

		HFACS-ML Level		
	iD	**Unsafe Supervision**	**Precondition**	**Unsafe Act**
Reference Doc	361	Inadequate Supervision	Personnel Factor	Error
Most Similar Docs (**D2V**)	771	Inadequate Supervision	Physical Env. 2	Error
	1191	Inadequate Supervision	Personnel Factor	Error
	1011	Inadequate Supervision	Personnel Factor	Error
	1411	Inadequate Supervision	Condition of Operator	Error
	857	Inadequate Supervision	Personnel Factor	Error
Score		5/5	3/5	5/5

Table 4: Least similar documents, from the D2V DM model.

		HFACS-ML Level		
	iD	**Unsafe Supervision**	**Precondition**	**Unsafe Act**
Reference Doc	361	Inadequate Supervision	Personnel Factor	Error
Least Similar Docs (**D2V**)	212	n/a	Physical Env. 2	n/a
	626	n/a	Technological Env.	n/a
	823	n/a	Technological Env.	n/a
Score		0/3	0/3	0/3

Note that in order to keep using cosine distance as the primary metric of vector similarity, all document projections have been normalized to unit length. Therefore, excluding magnitude from their differentiation.

4. Human Factor Label Propagation

During the last years, semi-supervised learning has emerged as an exciting new direction in machine learning research. It is closely related to profound issues of how to effectively infer from a small labelled set while leveraging properties of large unlabelled data. A challenge often found in real-world scenarios, where labelled data is expensive to acquire.

In this study, we analyse how the Label Spreading (LS) algorithm may propagate information to infer the intrinsic structure of the data and therefore predict human factors of unknown documents.

4.1. Label Spreading

Introduced in [20], this algorithm uses labelled nodes to interact as seeds which spread their information through the network, following an affinity matrix based on node distance and distribution. During each iteration, each node receives the information from its neighbours, while retaining a part of its initial information. The information is spread symmetrically until convergence is reached, and the label of each unlabelled point is converted to the class which it has received most information during the iteration process.

To define the affinity matrix, it may use a Gaussian Radial Basis Function (RBF) associated to a single hyper-parameter *Gamma*, which defines the weight with which two document vectors may influence each other. This process is shown in equation 6 and additional documentation regarding the algorithms can be found in [21].

$$K(d_2, d_2) = exp(-Gamma * ||d_1 - d_2||^2) \tag{6}$$

4.2. Evaluation Metrics

Multi-class classification metrics compare predicted results to ground truth labels not used during the training process. In this study, we established one primary metric, Micro F1 score, on which the models were optimized, and two other complementary metrics, Macro F1 score and Precision, that will be used to gain deeper insights into the results. Next, follows the expressions for each of the metrics.

$$Micro\,F1 = F1_{class1+...+classN} \tag{7}$$

$$Macro\,F1 = \frac{F1_{class1} + ... + F1_{classN}}{N} \tag{8}$$

$$Precision = \frac{\sum_{a \in A} TP_a}{\sum_{a \in A}(TP_a + FP_a)} \tag{9}$$

Where A is any set of main categories from a single level of the HFACS-ML framework. Note that complementary documentation of some of the used terms can be found in [22].

4.3. Early Findings

In an initial attempt to better understand how data extraction and prediction may be improved, an initial categorization experiment was globally carried out, using the baseline D2V DBoW embedding model together with the LS classifier.

For this experiment, we availed the previously labelled data and split it into train and test sets, in a stratified manner, over different train sizes (Ts). Figure 3 shows the confusion matrix appurtenant to the best result from the Precondition for Unsafe Act level, at $Ts = 0.36$.

It may be immediately noticed from Figure 3 that our multi-class classification system is largely affected by class imbalance. Because of this factor, especially evident for the exhibited level, we decided to down-sample the 'Physical Env. 1' category to an order of magnitude more similar to that of the other categories. The subsequent results are shown in Figure 4.

Through observation of the differences in the main diagonal, it is interesting to note, from Figure 4, that class balance and prediction evenness were considerably improved after down-sampling. Although the Micro F1 score remained roughly the same, around 0.54, the Macro F1 score increased from 0.25 to 0.34.

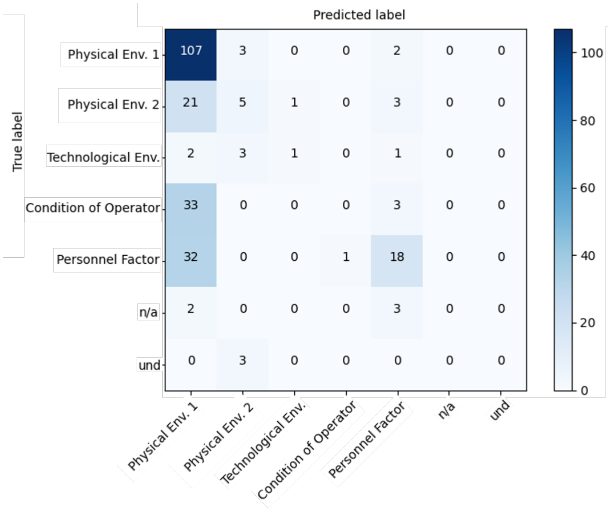

Figure 3: Precondition for Unsafe Act confusion matrix.

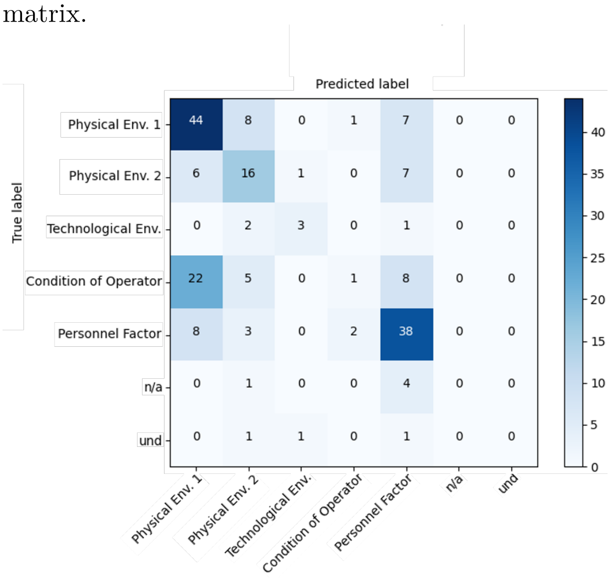

Figure 4: Precondition for Unsafe Act confusion matrix, with down-sampled ‘Physical Env. 1’.

Another observed irregularity, transversal to all levels of the framework, was the inefficiency of the outlier category ‘und’ to predict documents of the same class. Because it failed its purpose and contributed only to adding noise to the system, this category and the documents related to it were removed from the rest of the study.

4.4. Hyper-Parameter Impact Analysis

A procedure often followed by algorithm designers to improve model performance is hyper-parameter tuning. Yet, tuning complexity grows exponentially with the number of hyper-parameters and for certain scenarios, such as the present one, where this number is particularly large, a selection has to be made [23]. For this reason, we considered the functional Analysis of Variance (fANOVA) [24, 25] to help us narrow down which hyper-parameters account for the biggest impact on the objective function and therefore hold a higher need for tuning.

Since this approach requires the use of empirical data, we ran a random search with 350 different states, registering for each state the performance score (Micro F1) and the respective hyper-parameter configuration. The list of hyper-parameters, range, scale and type is shown in Table 5. Note that these trials were conducted on the Unsafe Act level. Due to being the most even of the framework, it was expected to provide the most reliable results.

Table 5: Random search characteristics of each hyper-parameter.

Hyper-Parameter	Min Value	Max Value	Scale	Type
Train size	0.2	0.8	Uniform	Float
Gamma	0.2	200	Log Uniform	Float
Dimensions	10	1000	Log Uniform	Integer
Window size	1	50	Log Uniform	Integer
Epochs	1	100	Log Uniform	Integer
Learning rate	0.0025	0.25	Log Uniform	Float
NS words	1	50	Log Uniform	Integer

After fitting our empirical data into the fANOVA process, we obtained the marginal contribution of each hyper-parameter (Figure 5). Note that the marginal contribution can be interpreted as the relative importance a certain variable over the final objective function.

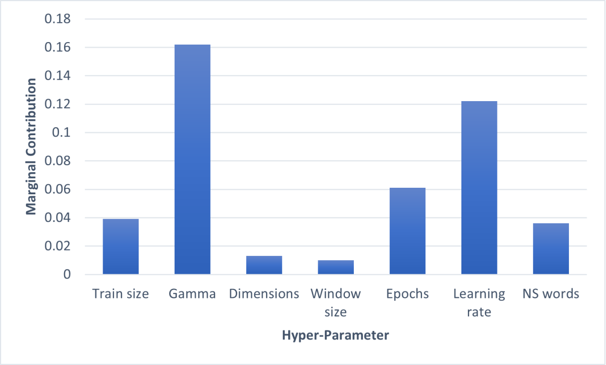

Figure 5: Marginal contribution of each hyper-parameter, predicted by the fANOVA.

From the bar plot exhibited in Figure 5, it may be observed that even in high-dimensional cases most performance variations are attributable to just a few hyper-parameters - in this case Gamma, Learning rate and Epochs - while others, such as Dimensions and Window size, seem to possess a much lower influence. These results are availed in the next subsection.

4.5. Bayesian Optimization

There exist a variety of industry-standard optimization approaches. In this work, we consider the automatic Bayesian optimization algorithm due to its ability to use previous objective function observations to determine the most probable optimal hyper-parameter combinations [26, 27]. This approach falls into a class of optimization algorithms called Sequential Model-Based Optimization (SMBO), and is capable of balancing exploitation versus exploration of the search space, for either sampling points which are expected to provide a higher score or regions of the configuration space which have not been explored yet.

With the aim of improving comprehension and steadily test the potentialities of Bayesian optimization, we applied this algorithm multiple times with an incremental number of free variables. For this implementation, we followed the order suggested by the fANOVA results (Figure 5), prioritizing hyper-parameters with higher marginal contributions. Note that the procedure shown in this subsection retracts again to the Unsafe Supervision level, but has been replicated for all levels of the framework.

Starting with Gamma, Figure 6 illustrates how the Bayesian optimization algorithm performed with one free variable, over a total of 100 iterations (shown on the left), and how it explores the relaxed state space (shown on the right).

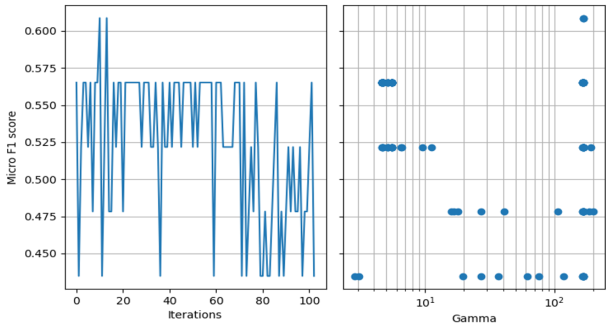

Figure 6: Bayesian optimization results with one free variable.

From Figure 6 it can be observed that although the optimization algorithm diligently explores different regions of the state space, the best achieved result of 0.61 is still not very good. To broaden the search scope, we ran the Bayesian optimization algorithm once again, but now with an additional free variable, Learning rate. Figure 7 shows the subsequent search distribution for both variables.

A far better result of 0.82 can be observed from this new configuration. We may also note from the data distribution that Gamma has been explored in some of the same regions as in the previous iteration, but with a drastically different outcome. This is an evident reflection of the high association between Gamma and Learning rate.

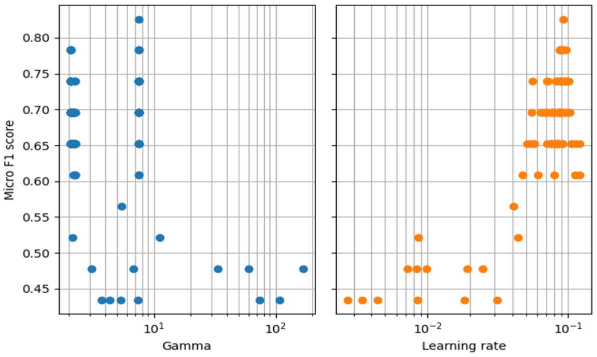

Figure 7: Bayesian optimization results with two free variables.

As the number of free variables increased, not necessarily did the final outcome. Similar values from the previous one were registered with three and four free variables, reaching a global best of 0.875 at four free variables. However, only lower values were obtained with five, six and seven free variables, reaching as low as 0.685 in these tests. This may suggest that, for the current non-convex maximization problem, there might be a limit to Bayesian optimization, situated around four free variables, which may be a consequence of the exponential expansion in configuration space, as well as in objective function complexity.

4.6. Metric Results

To test the effectiveness of the developed human factor classification algorithm, we took the best results from the Bayesian optimization models and compared them against other baseline embedding and classification techniques. For this, we tried the previously tested TF-IDF as a D2V substitute to represent the document vectors, and added a Support Vector Machine (SVM) as a potential substitute of LS for the task of vector classification. To make it a fair comparison, we also included the results from our non-optimized baseline model, D2V DBoW NS + LS, after ‘und’ removal.

Additionally, we took advantage of the random search infrastructure, initially built for the fANOVA process, and retrained all the embeddings on this search mechanism, therefore including another widely used optimization method in the analysis. The final comparison for each level of the framework is summarized in Tables 6, 7 and 8, respectively.

From the results observed in Tables 6, 7 and 8, we distinctively attribute the best performance to the Bayesian optimization approach, which exhibited much better results than the baseline model. Comparatively, random search provided acceptable results for a high enough number of iterations, but it did not prove to be as optimal or as consistent.

Table 6: Best predictions from the Unsafe Supervision level.

		Best Results		
Model Type	**Model Name**	**Micro F1**	**Precision**	**Macro F1**
Random Search	D2V DBoW NS' + LS	0.816	0.894	0.548
	D2V DBoW HS' + LS	0.850	0.928	0.531
	D2V DM NS' + LS	0.833	0.880	0.492
	D2V DM HS' + LS	0.800	0.900	0.471
Bayesian Optimization	D2V DBoW NS" + LS	0.900	0.933	0.578
	D2V DBoW HS" + LS	0.900	0.933	0.578
	D2V DM NS" + LS	0.850	0.928	0.510
	D2V DM HS" + LS	0.850	0.866	0.518
Baseline Models	D2V DBoW NS + LS	0.800	0.750	0.488
	D2V DBoW NS + SVM	0.500	0.500	0.133
	TD-IDF + LS	0.650	0.625	0.378
	TF-IDF + SVM	0.600	0.555	0.276

Table 7: Best predictions from the Precondition for Unsafe Act level.

		Best Results		
Model Type	**Model Name**	**Micro F1**	**Precision**	**Macro F1**
Random Search	D2V DBoW NS' + LS	0.657	0.656	0.659
	D2V DBoW HS' + LS	0.729	0.732	0.782
	D2V DM NS' + LS	0.610	0.618	0.517
	D2V DM HS' + LS	0.573	0.573	0.375
Bayesian Optimization	D2V DBoW NS' + LS	0.729	0.724	0.693
	D2V DBoW HS" + LS	0.779	0.789	0.735
	D2V DM NS" + LS	0.644	0.644	0.536
	D2V DM HS" + LS	0.627	0.632	0.494
Baseline Models	D2V DBoW NS + LS	0.407	0.407	0.197
	D2V DBoW NS + SVM	0.441	0.441	0.185
	TD-IDF + LS	0.763	0.759	0.745
	TF-IDF + SVM	0.661	0.661	0.560

Table 8: Best predictions from the Unsafe act level.

		Best Results		
Model Type	**Model Name**	**Micro F1**	**Precision**	**Macro F1**
Random Search	D2V DBoW NS' + LS	0.800	0.800	0.818
	D2V DBoW HS' + LS	0.731	0.710	0.735
	D2V DM NS' + LS	0.704	0.704	0.495
	D2V DM HS' + LS	0.741	0.739	0.761
Bayesian Optimization	D2V DBoW NS' + LS	0.875	0.923	0.859
	D2V DBoW HS" + LS	0.826	0.842	0.830
	D2V DM NS" + LS	0.782	0.800	0.755
	D2V DM HS" + LS	0.869	0.850	0.898
Baseline Models	D2V DBoW NS + LS	0.565	0.526	0.560
	D2V DBoW NS + SVM	0.522	0.522	0.288
	TD-IDF + LS	0.739	0.737	0.762
	TF-IDF + SVM	0.652	0.652	0.447

As for the comparison between models, various conclusions may be extracted. In a primary analysis, it can be observed that the DBoW architecture generally performed slightly better than the DM for the current data set. In a second inspection, it can also be observed that the supervised SVM did not perform as well against class imbalance, presenting always the lowest Macro F1 scores. In contrast, a surprisingly good result came from the baseline TF-IDF + LS model, significantly surpassing the baseline D2V DBoW NS + LS on two levels of the framework. Due to this result, we also explored optimizing this model. However, it did not surpass the best results, described in Tables 6, 7 and 8, for any of the experiments.

5. Conclusions and Future Work

In this section, we summarize the results obtained from this study and propose some ideas for future work.

5.1. Conclusions

The results obtained in this study showed that the semi-supervised LS algorithm was an appropriate classifier for the current setting, particularly for the HFACS-ML levels which possessed fewer labels. We do not discard the potential of the supervised SVM, for the same purpose, but note that it might prove more reliable for larger and more even labelled data sets. Surprisingly, the TF-IDF model was also observed to be an interesting alternative to D2V, for some levels of the framework, although it also proved to be more computational expensive due to its high dimensionality.

A final relevant conclusion to be taken from this study is the usefulness of Bayesian optimization, when properly tuned, for finding near-optimal hyper-parameter combinations over non-convex objective functions. The fANOVA marginal contribution analysis was also crucial for this purpose, providing valuable insight into the most influential hyper-parameters.

5.2. Future Work

In the present dissertation, a novel HFACS-ML framework has been proposed. It would be interesting to perform a study comparing how it would stack against the original HFACS, on the same task. It could also be pertinent to investigate how different variations from these frameworks could better fit other machine learning applications and data sets.

Another concept that should also be considered is the inclusion of a larger labelled and unlabelled data set, in order to understand how this work could perform in a scaled scenario. This fact also motivates further research regarding other approaches for constructing labelled data sets. An interesting alternative to the methods used here is Active Learning, a methodology that prioritizes the labelling of uncertain points, instead of randomly selected documents, so as to optimize convergence of label propagation algorithms.

Finally, feature selection analysis, such as redundancy and noise, should be carried out in greater depth. In the particular case of the developed models, this is a very important topic, since these operate based on the quality and size of the vocabulary. More work can also be done around exploring other types of feature extraction and classification algorithms, as well as their respective combinations.

References

[1] K. A. Latorella and P. V. Prabhu. A review of human error in aviation maintenance and inspection. 26(2):133–161.

[2] F. Schreiber. Human performance - error management.

[3] National Research Council. *Improving the Continued Airworthiness of civil aircraft: A strategy for the FAA's Aircraft Certification Service*. The National Academies Press.

[4] ICAO. The world of air transport in 2018.

[5] Y.-L. Hsiao, C. Drury, C. Wu, and V. Paquet. Predictive models of safety based on audit findings: Part 1: Model development and reliability. 44(2):261–273.

[6] Y.-L. Hsiao, C. Drury, C. Wu, and V. Paquet. Predictive models of safety based on audit findings: Part 2: Measurement of model validity. 44(4):659–666.

[7] T. Pham, T. Tran, D. Phung, and S. Venkatesh. Predicting healthcare trajectories from medical records: A deep learning approach. 69:218–229.

[8] J. Liu, Z. Zhang, and N. Razavian. Deep EHR: Chronic disease prediction using medical notes. In *Proceedings of the 3rd Machine Learning for Healthcare Conference*, volume 85 of *Proceedings of Machine Learning Research*, pages 440–464. PMLR.

[9] E. Masciari, V. Moscato, A. Picariello, and G. Sperli. A deep learning approach to fake news detection. In D. Helic, G. Leitner, M. Stettinger, A. Felfernig, and Z. W. Raś, editors, *Foundations of Intelligent Systems*, volume 12117 of *Lecture Notes in Computer Science*, pages 113–122. Springer.

[10] Ludovic Tanguy, Nikola Tulechki, Assaf Urieli, Eric Hermann, and Céline Raynal. Natural language processing for aviation safety reports: From classification to interactive analysis. *Computers in Industry*, 78:80 – 95, 2016. Natural Language Processing and Text Analytics in Industry.

[11] ASN. ASN aviation safety database.

[12] D. A. Wiegmann and S. A. Shappell. *A Human Error Approach to Aviation accident analysis. The human factors analysis and classification system*. Ashgate Publishing Limited.

[13] A. K. Uysal and S. Gunal. The impact of preprocessing on text classification. 50(1):104–112.

[14] S. Vajjala, B. Majumder, A. Gupta, and H. Surana. *Practical Natural Language Processing: A comprehensive guide to building real-world NLP systems*. O'Reilly Media.

[15] S. Bird. Nltk 3.5.

[16] K. Spärk Jones. A statistical interpretation of term specificity and its application in retrieval. In P. Willett, editor, *Document retrieval systems*, pages 132–142. Taylor Graham Publishing.

[17] M. Kusner, Y. Sun, N. Kolkin, and K. Weinberger. From word embeddings to document distances. In *Proceedings of the 32nd Int. Conf. on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 957–966. PMLR.

[18] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In Y. Bengio and Y. LeCun, editors, *Workshop Proceedings of the 1st Int. Conf. on Learning Representations*, pages 1–12.

[19] Q. V. Le and Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st Int. Conf. on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196. PMLR.

[20] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems*, volume 16, pages 321–328. The MIT Press.

[21] Scikit learn developers. Sklearn.semi_supervised.LabelSpreading.

[22] S. Godbole and S. Sarawagi. Discriminative methods for multi-labeled classification. In H. Dai, R. Srikant, and C. Zhang, editors, *Advances in Knowledge Discovery and Data Mining*, volume 3056 of *Lecture Notes in Computer Science*, pages 22–30. Springer.

[23] F. Hutter, H. Hoos, and K. Leyton-Brown. An efficient approach for assessing hyperparameter importance. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st Int. Conf. on Machine Learning*, volume 32(1) of *Proceedings of Machine Learning Research*, pages 754–762. PMLR.

[24] G. Hooker. Generalized functional anova diagnostics for high-dimensional functions of dependent variables. 16(3):709–732.

[25] F. Hutter and S. Falkner. fANOVA 2.0.5 documentation.

[26] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The bayesian optimization algorithm. In W. Banzhaf, J. M. Daida, A. E. Eiben, M. H. Garzon, and V. Honavar, editors, *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation*, volume 1, pages 525–532. Morgan Kaufmann Publishers.

[27] J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 2951–2959. Curran Associates.