12-2021

Identifying Incident Casual Factors to Improve Aviation Transportation Safety: Proposing a Deep Learning Approach

Tianxi Dong
Trinity University, tdong@trinity.edu

Q. Yang

N. Ebadi

X. R. Luo

P. Rad

Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5540046, 15 pages
https://doi.org/10.1155/2021/5540046

Research Article

Identifying Incident Causal Factors to Improve Aviation Transportation Safety: Proposing a Deep Learning Approach

Tianxi Dong,[1] **Qiwei Yang**,[2] **Nima Ebadi**,[2] **Xin Robert Luo**,[3] **and Paul Rad**[4]

[1]*School of Business, Trinity University, One Trinity Place, San Antonio, TX 78212, USA*
[2]*Department of Electrical and Computer Engineering, The University of Texas, San Antonio, TX 78249, USA*
[3]*Anderson School of Management, The University of New Mexico, Albuquerque, NM 87131, USA*
[4]*Department of Information Systems and Cyber Security, The University of Texas, San Antonio, TX 78249,, USA*

Correspondence should be addressed to Tianxi Dong; tdong@trinity.edu

Received 22 January 2021; Revised 24 March 2021; Accepted 25 May 2021; Published 14 June 2021

Academic Editor: Chi-Hua Chen

Aviation is a complicated transportation system, and safety is of paramount importance because aircraft failure often involves casualties. Prevention is clearly the best strategy for aviation transportation safety. Learning from past incident data to prevent potential accidents from happening has proved to be a successful approach. To prevent potential safety hazards and make effective prevention plans, aviation safety experts identify primary and contributing factors from incident reports. However, safety experts' review processes have become prohibitively expensive nowadays. The number of incident reports is increasing rapidly due to the acceleration of advances in information technologies and the growth of the commercial and private aviation transportation industries. Consequently, advanced text mining algorithms should be applied to help aviation safety experts facilitate the process of incident data extraction. This paper focuses on constructing deep-learning-based models to identify causal factors from incident reports. First, we prepare the data sets used for training, validation, and testing with approximately 200,000 qualified incident reports from the Aviation Safety Reporting System (ASRS). Then, we take an open-source natural language model, which is well trained with a large corpus of Wikipedia texts, as the baseline and fine-tune it with the texts in incident reports to make it more suited to our specific research task. Finally, we build and train an attention-based long short-term memory (LSTM) model to identify primary and contributing factors in each incident report. The solution we propose has multilabel capability and is automated and customizable, and it is more accurate and adaptable than traditional machine learning methods in extant research. This novel application of deep learning algorithms to the incident reporting system can efficiently improve aviation safety.

1. Introduction

In the last two decades, we have witnessed rapidly evolving customer expectations and paradigmatic business mergers and acquisitions in the mushrooming development of the aviation industry. In this highly competitive environment, airline companies have increasingly exploited information technologies to turn challenges into business opportunities and support decision-making. Automated decision support technologies remain one of the main challenges in air transportation [1]. Aviation incident reporting and investigation systems are a crucial part of the ongoing digitization of safety efforts. Incidents are anything abnormal that affects or could affect the safety of aviation operations [2]. Unlike accidents, which usually involve fatalities or serious injuries, incidents are much more frequent and less costly than accidents. They are a valuable source of data to help identify potential hazards. Incident reports record various abnormal events and provide reference data to the Federal Aviation Administration, the National Aeronautics and Space Administration, and the National Transportation Safety Board, during the processes of decision-making, procedure design, threat identification, training, and so forth [3]. Since aviation transportation is a highly sophisticated system, many factors, such as human error, aircraft mechanical failure, extreme weather, and unreasonable

company policy, or a combination of them, can result in incidents. Due to the paramount value of incident data, countries and multinational institutes have devoted significant efforts to collecting and storing incident reports for analytical decision-making.

The Aviation Safety Reporting System (ASRS), jointly operated by the FAA and NASA, is one of the leading aviation incident reporting systems and is used extensively in North America. The system receives aviation incident reports submitted by airports, airline companies, pilots, and crews daily. Then the system analyzes and responds to incident reports to identify potential hazards early and prevent aviation accidents. Incident reporting and investigation systems are critical components of safety management in aviation transportation [4]. The information frequently encountered in incident investigations includes the events leading up to the accident, the factors that increased risk, the detection of problems, and the attempts to resolve the problems, all of which can be provided by individuals involved in incidents [5]. The ASRS, a rich and reliable database of information on aviation incidents, is used by NASA and the FAA to evaluate the effectiveness of risk management actions. As a distinctive contribution to safety management, the feedback from incident reporting systems is a vital early-warning tool for decision-makers and planners tasked with improving safety margins in the face of doubled or quadrupled operations [4].

Most of the incident reports are submitted to the ASRS voluntarily. A reporter involved in an incident can fill out an ASRS reporting form anonymously. The narrative is the most informative part of an incident report. The reporter recounts the actual events before, during, and after the incident. Narrative texts mostly describe mechanical failures, observations, behaviors, and weather conditions related to the incident. All submitted ASRS reports are currently manually analyzed and assigned at least one out of sixteen primary factors and no more than four out of sixteen contributing factors by experienced aviation safety analysts [6]. The identification of the primary and contributing factors is a crucial step. The tabular data collected from the reporter includes 96 tabular attributes, such as the reporter's role, qualifications, and experience, type of aircraft involved, type of operator, cabin activity, weather, and many other event-specific details. Unfortunately, based on a random selection of 10,000 incident reports, more than 50% of the incident reports are missing at least half of these attributes, and most of the attributes that are often present, such as date, local time, and state, seem to have little relevance to the causes of the incidents. Thus, the current predicament is that each incident report's narrative text data is the only reliable and informative source to identify the incident-causing factors. Table 1 is an example of a typical ASRS incident report and the conclusions made by human experts (Tables 1 and 2).

The analysis of incident causal factors in the incident reports has been helpful in investigating the root causes of aviation incidents. The research conducted in [7] studied design-induced problems in Flight Management Systems (FMSs) by selecting 99 incident reports related to FMSs from the ASRS. It concluded that a significant number of operational and design-induced problems exist in FMSs, because the user interface of FMSs is not optimally designed. Manufacturers should find a better balance in FMS design between logic and ease of use to reduce the occurrence of errors. Another study [8] used 37 incident reports from the National Transportation Safety Board (NTSB) database to study errors in decision-making in the aviation domain and discussed the nature of such errors, what main factors contribute to them, and what solutions might mitigate them. Reference [9] analyzed the causal factors in aviation maintenance by investigating 3,783 ASRS incident reports related to maintenance incidents. It concluded that individual-related and management-related factors are the most frequent reasons for maintenance error. The nonmaintenance perspective should be given more attention because it can provide abundant information that is usually not included in maintenance personnel reports. To study the multifactor and single-factor effects on human performance in Air Traffic Management (ATM), [10] used over 400 European aviation incident reports related to ATM as their source data. The research concluded that research focusing on single-factor (stress, fatigue, communication, etc.) effects on human performance is poorly suited to the complexities of contemporary ATM, because incident reports often indicated multifactor cooccurrences. In sum, a collection of aviation safety research and analysis has relied on incident reports and their conclusions about causal factors. At present, the ASRS heavily depends on human experts to identify the causal factors. However, the increasing number of incident reports submitted every day, due to the rapid growth of the aviation industry, has caused analysis of the newly generated incident reports to be delayed by three to six months. This delay reduces the effectiveness of the ASRS as an early-warning system for decision-makers, aviation organizations, and government agencies.

The situation described above has become increasingly urgent in recent years due to the burgeoning growth of commercial air transportation, private aircraft, and unmanned aircraft systems in the aviation industry [1], thereby yielding a quickly mounting number of incident reports. Figure 1 shows annual incident reports ASRS received over the last 28 years. For instance, ASRS only received approximately 4,600 incident reports in 1981, compared with about 108,000 incident reports per month in 2019. Worse yet, the lack of timely and accurate analysis of the incident reports substantially reduces the value of the data, making effective safety prevention and improvement strategies increasingly challenging (Figure 1).

Safety in aviation transportation is crucial. Analyzing incident reports quickly and accurately on a large scale facilitates the decision-making process and makes early detection and prevention of potential hazards possible. In this study, we build a deep-learning model that can identify not only *primary factors* but also *contributing factors* with promising results described later on. The main contributions of our research to reduce gaps in extant research are summarized as follows:

Table 1: The example of an incident report and its analysis results.

Incident report submitted:
Narrative: busy session numerous over flights requiring course changes to avoid traffic. Six or seven arrivals to different airports descending through over flights and several departures. A satellite propeller arrival was coming in from the north at 10000 and an over flight was off the departure end NE bound at 11000 two F16 south departed routed north climbing to 15000 a military intercept was squawking 7777. I assumed this was in error and I informed the lead. I turned the aircraft to 020 heading to split the other traffic and allow the climb to continue. I consciously thought about re-assigning the altitude but after the squawking the turn and traffic issuance, I did not want to throw any more numbers at the pilot who would increase the transmissions and potential confusion. I wanted to get this guy on course and off my frequency but had to wait until he topped. The guy at 10000 made numerous other transmissions. Then looked at the F16 south and he was climbing very fast as I was about to transmit the F16 south showed 16000 and asked intermediate fix. They were cleared to the block. I said no, assigned altitude 15000, contact ZKC. No traffic observed, but at 560 knots aircraft can mess things up pretty quick.
Tabular data:

96 attributes {
- Time: 200905
- Local Time of Day: 1201 – 1800
- Relative Position: (missing)
- ⋮
- ⋮
- Cabin Light: (missing)

Results analyzed by human experts:
Primary Factor: human factors
Contributing Factors: human factors, procedure, aircraft
Synopsis: arbus flight crew landing on runway 8L at ATL reports a runway incursion after being instructed to taxi via delta; bravo; Victor; Foxtrot to the ramp. Crew failed to turn on to bravo and entered 8R at delta. An EMB170 crew had to reject their takeoff on runway 8R.

Originally, an incident report comprises two components, *Narrative* and *Tabular data*. In most cases, *Tabular data* is neither reliable nor useful because it is either missing or not quite related to the incident. After being reviewed by human experts, *Primary Factor*, *Contributing Factors*, and *Synopsis* are the conclusions generated from this incident.

Table 2: A comparison of our study with extant research related to aviation reporting system.

Studies	Research target	Data set	Algorithms	Performance
Tiller et al. [14]	Analyze close call incident reports to assess severity level	117 reports from the ASRS (2014–2016)	Bliss's taxonomy, a manual case-by-case review process	Modification on the close call taxonomy is needed, but results were not discussed quantitatively
Tanguy et al. [2]	Extract metadata and keywords from the narratives, and topic mining	86,912 qualified reports used from DGAC	N-Grams Support Vector Machine topic modeling	Incident reports classified to seven major topics, with about 78% F_1 score on average
Kuhn [15]	Automate the topic mining process	ASRS incidents from 2010 to 2015 (the exact number is not specified)	N-Grams topic modeling	Some incidents are closely related to key words, and topic modeling identified those well, but results were not evaluated quantitatively
Robinson [13]	Identify the contributing factors of the incident reports	7,484 incident reports from the ASRS	Latent semantic analysis	Identify the multiple factors of each incident; the accuracy needs significant improvement
Shi et al. [4]	Identify two primary causal factors of incidents with machine learning	168,227 incidents from the ASRS	Naive Bayes Hoeffding tree OzaBagADWIN	Automate to identify two most casual factors, and topic mining used to extract structured information
Our study	Identify the primary factor and multiple contributing factors of each incident from six most causal factors	181,651 incident reports from the ASRS	Deep recurrent neural networks	Demonstrate that deep learning is a powerful tool for processing complex textual data. We achieve best performance so far to identify the primary factor and contributing factors among related research

(1) Rather than directly addressing the task of classifying incident reports, we make an early attempt to introduce a well-trained deep-learning language baseline model that can "understand" general English texts, and then we refine our model based on the performance of the baseline model to cope with the incident reports. Our research shows that about 4% accuracy is gained.

(2) To the best of our knowledge, our study is the first attempt to perform a multiclass and multilabel operation on ASRS incident reports on a large scale. Our study pushes the application of deep learning methods in the safety management domain forward. We propose suitable metrics to evaluate the performance of this multiclass and multilabel classification, which is rarely used in extant research as they

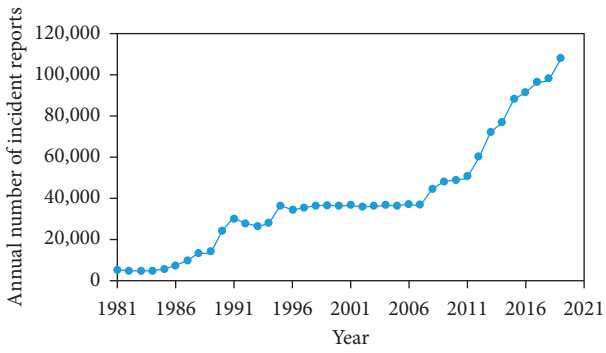

Figure 1: Annual incident reports ASRS received from 1981 to 2019.

primarily focus on binary or single-label classification.

(3) Our study demonstrates the high adaptability and reusability of deep learning methods. Therefore, our proposed deep learning methods are applicable to many tasks that demand text analysis, especially in an automated way. In addition, once the data is updated or the task is changed somewhat, the developed deep learning model can be modified accordingly without starting over from scratch.

This study establishes a fruitful research foundation for researchers who seek to apply deep learning methods to the solution of a myriad of text analysis problems in general and especially for those whose corpora include a customized vocabulary of technical terms. Our proposed approach sheds light on nontrivial optimizations to improve the baseline model's accuracy, as we strive to present a procedure to develop a deep learning model to help solve the pressing problem of aviation safety decision support.

The rest of this paper unfolds as follows. Section 2 is a review of relevant research. In Section 3, we describe the raw data and statistics and how to prepare them to be suitable for the training in the next step. Section 4 briefly introduces the main steps to build a deep recurrent network model using Python deep learning libraries and refine it based on our specific task. Section 5 epitomizes the experiments to determine hyperparameters in the model. We highlight the critical parameters that often significantly affect the performance of deep learning models, and we introduce new metrics to evaluate the results and compare them with related extant research. Section 6 discusses the potential implications of our research, and Section 7 presents the conclusions and limitations of this study.

2. Related Work

2.1. Automated Incident Analysis in Safety Management. Safety management is a continuous improvement process that reduces hazards and prevents incidents in aviation. The incident reporting system is a crucial part of safety management, as it collects data and evidence for decision-making, identifies potential risks to help prevent accidents, and provides examples to educate personnel. Extant research primarily concentrates on text mining techniques to automate the analysis of incident reports. Therefore, extant research has attempted to apply machine learning techniques to extract textual information. Table 2 compares this study with extant research that used aviation incident data. Tixier et al. [11] examined 2,200 construction incident reports by applying a rule-based automated content analysis system. The length of the sample reports presented in their paper was usually less than 50 words, and they primarily manually mapped keywords to specific incidents. Therefore, their proposed method is not easily applicable to lengthy and complicated narratives. Mousa et al. [12] proposed the XGboost algorithm to classify 13,165 highway-railroad crossing incidents and reported an accuracy of 99.11%. However, other baseline methods, such as Decision Tree or Random Forest, also achieve around 98.5% accuracy. Therefore, it is likely that the incident reports they were dealing with are naturally easy to differentiate. Shi et al. [4] applied manual feature engineering to the ASRS data set with Term Frequency-Inverse Document Frequency (TF-IDF) and fed the features into three supervised machine learning algorithms, Naive Bayes, Random Forest, and Support Vector Machine (SVM), to identify the two most frequent primary factors: "human factors" and "aircraft." The shortcomings of this research are that primary factors, "human factors," and "aircraft" combined account for about 81% of all incidents, and, even with only the two most frequent primary factors selected, the three traditional machine learning methods used in the research could only achieve an average accuracy about 81% at best. Therefore, a practical model that can handle more factors with improved accuracy is needed. Tanguy et al. [2] built classifiers with French national aviation occurrence data (DGAC[1]). The authors employed manual feature engineering using N-Grams and topic modeling and used the extracted features to train an SVM classifier. Rather than attempting to identify the primary factors from the incident reports, their goal was to discover the main topics of the incident, such as "cabin," "ground," and "weather." The disadvantage of their method is that, even when things like "cabin" and "weather" are mentioned in an incident report, they are not necessarily the actual factors that caused the incident. Robinson [13] was one of the first authors to tackle multilabel classification using an ASRS data set. The author built a latent semantic analysis (LSA) model, trained it with 4,497 incidents, and tested the model on 2,987 other incidents. However, the author reported poor model performance with an average F_1 score of 0.409 due to the small sample size used in the research overly ambitious attempt to classify all factors.

Our literature review indicates research gaps existing in the extant research. Most of the extant studies only use a relatively small number of data samples to develop their models. Models developed in this way may only be applicable to limited data sets. However, transportation incident reports are usually highly unstructured. Furthermore, although Shi et al. [4] used an extensive data set in their research, they only addressed the two most frequent factors, human factors and aircraft, which account for about 80% of all incidents, and ignored the rest. Such oversimplification

restricts the model to limited applications. The proposed methods in extant research are subject to two significant shortcomings: (1) a lack of high accuracy (less than 80%) and (2) a limited number of primary and contributing factors. Therefore, effectively automated identification of multiple incident factors to support decision-making remains one of the main challenges in aviation reporting systems. Due to various contributing factors such as human factors, aircraft, weather, and company policy [16], the inherent complexity of aviation operations requires reviewers with aviation experience to make sensible judgments. Accumulated evidence of the successful application of deep learning methods to the analysis of incident reports could bring about the acceptance of this approach as a solution to aviation safety management.

2.2. Emerging Deep Learning Methods in Transportation. In the last few years, deep recurrent networks, a subclass of deep learning methods, have been widely applied in transportation decision-making systems and have achieved promising results. Dong et al. [17] applied deep neural networks to predict traffic crashes. The study shows the advantages of deep learning methods over SVM, including automatic feature extraction, superior performance, and the ability to handle heterogeneous data. Cortez et al. [18] used bidirectional long short-term memory (LSTM) to predict emergency events using data from the Korean Ministry of the Interior in 2015, and the LSTM model showed better performance than SVM and time series models. A more recent aviation study [19] used recurrent networks to predict flight trajectory and their results illustrated the promising performance of the blended deep learning model in predicting flight trajectory and assessing en-route flight safety. Luo et al. [20] combined KNN and LSTM to predict traffic flow. KNN was used to address spatial data and LSTM for temporal data. The study reported that the deep learning method achieved superior performance on real traffic data. All the above studies have successfully shown the superiority of deep learning methods on large and unstructured data sets over traditional machine learning algorithms.

The deep neural network model, which combines the advantages of unsupervised and supervised learning algorithms, is superior to traditional machine learning algorithms in many respects, especially in this "Big Data" era. Instead of the manual feature engineering required by traditional machine learning algorithms, deep learning methods can extract intrinsic features without human intervention. The manual feature engineering is primarily based on word frequency statistics [21], such as TF-IDF and N-Grams. Its main shortcoming is that it has difficulty in capturing the relationships among textual data accurately. In deep neural networks, on the other side, the word is represented as a high-dimensional vector using a skip-gram technique [22]. In this way, intrinsic relationships among words and the meaning of each word can be constructed and calculated, and this approach has yielded outstanding results [23]. Second, another advantage of deep neural networks is that traditional machine learning methods primarily predict by merely counting the word frequencies or probabilities of words that appear together, rather than extracting the meaning of the word based on its semantic context. However, deep neural networks have the ability to "remember" or store previous information. This ability is beneficial for building relationships among words that do not appear close to each other. This ability is crucial to our tasks because incident reports may not be written in an organized and concise way. That is one of the main reasons why the automatic analysis of incident reports is challenging. Last, deep neural networks are naturally suitable for use with a large amount of textual data. More data is helpful to refine the word embeddings [24]. Word embeddings are also called word vectors. They are a way of converting textual data *o* numbers. Unlike other common ways of embedding, such as frequency embedding, TF-IDF, Count Vectors, and word vectors are initialized randomly, then trained, and refined with a large corpus of texts. The essence of word embedding is that all the other words in the context decide the value of a word vector. Mikolov et al. [25] developed this method, and it has gained significant attention in natural language processing since then. With word embeddings applied, the model can evolve along with the accumulation of incident reports, as the ASRS is constantly receiving them.

Despite being powerful and efficient type of algorithms successfully applied to many domains, deep learning methods have found limited implementation in transportation incident reporting systems, which require natural language processing. The goal of this paper is to cover this research gap by building deep recurrent neural networks that can automate aviation incident report analysis with better performance than extant research.

3. Data Preparation

3.1. Data Descriptive. We downloaded about 200,000 incident reports from the ASRS database ranging from January 1988 to July 2020 when accessed on October 2, 2020, yielding a total of 181,651 qualified reports. Other unqualified reports, such as those without labels or those that are too short (fewer than 20 words), are discarded. Every incident report is composed of four pieces of text from two persons (their narratives and callbacks), which we have combined as a single narrative text sent to our model. Figure 2 shows the distribution of the number of words and sentences in our data sets. The considerable variations of number of words and sentences make it more difficult to build a robust model.

There are 16 primary factors identified by human experts in aviation incidents; however, we only use incident reports involving the six most frequent categories of *human factors* (HF), *aircraft* (AC), *company policy* (CP), *procedure* (PR), *weather* (WE), and *airport* (AP), which make up 95% of the incident reports. Incidents attributed to rare factors are not considered in this research, because they only account for a fraction of all incidents and would need more data to generate meaningful results. We believe that our research thus achieves a reasonable balance in terms of performance, feasibility, and reasonable simplification. Table 3 lists all primary factors and their percentages of all incidents. The highlighted factors are used in this study and other rare factors are ignored.

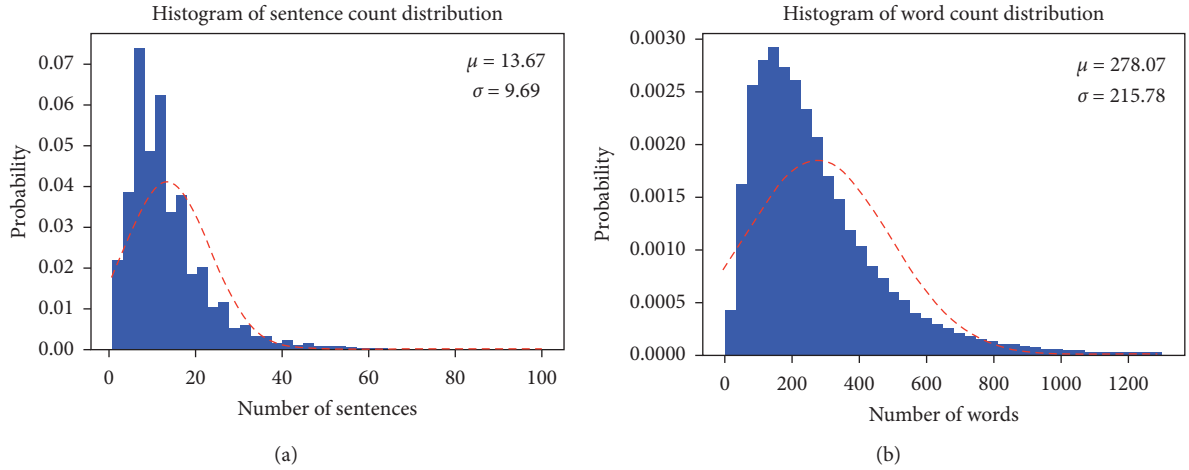

Figure 2: Distributions of lengths of incident reports based on the number of sentences and words.

Table 3: There are sixteen primary causal factors identified by the human experts in the database.

Primary casual factor	Count	Percentage in all incidents
Human factors	112,305	58.6%
Aircraft	43,119	22.5%
Company policy	7,676	4.0%
Procedure	7,626	4.0%
Weather	6,450	3.4%
Airport	4,475	2.3%
ATC equipment/buildings	2,803	1.5%
Chart or publication	2,519	1.3%
Environment, non-weather-related	2,180	1.1%
Airspace structure	1,163	0.6%
Equipment/tooling	465	0.2%
Manuals	338	0.2%
Staffing	238	0.1%
MEL	211	0.1%
Incorrect/unavailable part	154	0.1%
Logbook entry	32	0

The distribution of causal factors is highly unbalanced. Extant research primarily focus on the identification of the first two factors and ignore others. This study addresses the six most frequent factors, which account for as much as 95% of all incidents. Therefore, our solution is more applicable and feasible, because it can handle more factors, and is not targeting all factors, which causes the prediction performance to be worse due to the data unbalance.

In this research, we use narrative texts as the input to our model and, according to the input, our model predicts the primary (single-label) and contributing factors (multilabel) and compares them with the actual labels to evaluate the model's performance. We do not use the "Synopsis" section of each report as an extra input, because it is not the original content of the incident report and would make our automated text analysis less convincing.

Table 4 summarizes the essential statistics about multiple causal factors in ASRS data sets. Factor (or label) cardinality [26, 27] indicates that there are 1.47 factors (1 primary and 0.47 contributing factors) per report on average across all incident reports. This is the underlying reason for our decision to train our model to predict up to two factors for a single incident report, as mentioned in section 2. Identifying more than two factors for each incident report is not necessary in our research because cases of more than two factors are rare, and it would introduce unnecessary complexity without obvious performance gain. There are 28 distinct causal factor sets cooccurring in all incident reports, of which the most frequent combination is that of human factors and aircraft.

Table 5 shows the distribution of the six most frequent causal factors in detail. The overall occurrence of *human factors* (HF) is over 26 times more than that of *airport* (AP). The imbalance of the data distribution is likely to cause the classifiers to be biased toward the dominant category, in this case, human factors. Oversampling is applied to augment rare samples to overcome this issue. The other method we use to mitigate the bias is to apply a confidence threshold to *human factors*. Both are discussed in Section 5.

3.2. Data Preprocessing. We preprocess the narrative texts to reduce complexity and make the model more robust. Initially, the words in the report are tokenized into a list of its constituent words. Punctuation and stop words are removed in this step as they are not useful for text analysis [28]. Stemming and lemmatization are also applied to the input to decrease the number of distinct words and consequently reduce the model's complexity. To perform stemming and lemmatization accurately, a recognized Python library, the Natural Language Toolkit (NLTK) [29], is utilized. The ASRS extensively uses 537 acronyms for the words and phrases that frequently appear in narratives to make raw texts concise. For example, "STOL" stands for "Short Takeoff and Landing," and "VLF" represents "Very Low Frequency." These acronyms are decoded to their full words as the word vectors of acronyms are not seen in the pretrained word embeddings, which has been trained with the Wikipedia corpus. In addition, there are many meaningless words (or noise) existing in the corpus, such as "eeegl3," "shedcb," and "sewart." Thus, we remove any word that appears fewer than four times in our ASRS data sets. The study

Table 4: Important statistics about the utilized ASRS data set.

Multilabel statistics	Value
Number of utilized factors	6
Number of valid samples	172,990
Factor cardinality	1.47
Factor density	0.245
Number of distinct label sets	28
Most frequent label set	{Human factor, aircraft}

After cleaning and preprocessing, we use the six most frequent labels from 172,990 reports. On average, every report has 1.47 labels (label density of 0.245).

Table 5: The distribution of the number of labels along with distribution of labels within each number.

Number of labels	Total (%)	HF (%)	AC (%)	CP (%)	PR (%)	WE (%)	AP (%)
One	65.3	42.7	17.4	0.9	2	0.8	1.3
Two	24.3	14.6	4.6	2.01	1.05	1.16	0.8
Three	8.7	4.88	1.38	0.73	0.47	1.01	0.25
Four	1.6	0.8	0.2	0.15	0.08	0.3	0.08
Overall	100	63.0	23.6	3.8	3.6	3.3	2.4

The data is interpreted in this way; take the highlighted number for instance; 14.6% of all incident reports are marked exactly two causal factors (labels), and one of them is *HF*. It shows that *HF* prevails in both single and multiple labels.

[30] also used this straightforward but effective method to remove uncommon and useless words. In this way, many uncommon words are removed, while the important information of each incident report is kept intact. After preprocessing, a total of 6,960 unique words remain from 181,651 incident reports in this study.

As shown in Table 5, the distribution of the incident categories is highly imbalanced. Oversampling is used to augment the original data, because removing data from overrepresented classes, called undersampling, would not have been conducive to our deep learning approach, as deep learning improves with more data. Oversampling is a process that augments the data samples of underrepresented classes by copying them a certain number of times. In this study, incident reports labeled "aircraft" are copied two times, and those labeled "airport" ten times, and they are put back in the training data set. Finally, as shown in Table 6, of 181,651 incident reports, 80% are randomly picked as the training data set, 10% are used as the validation data set, and 10% are reserved as test data to measure model performance [31]. We apply oversampling after splitting the data to avoid data leakage between training, validation, and test sets. Unlike the validation data used by the model to monitor its performance during the training process, test data is kept isolated until the evaluation stage to guarantee the validity of the test data sets.

In this study, we only use oversampling to augment training data sets to identify primary factors. Regarding contributing factors, there is no noticeable performance gain from oversampling according to our experiments, because contributing factors are already mixed.

4. Methodology

4.1. Analysis and Processing of Aviation Incident Reports. The aviation incident reports are primarily free-form text describing each incident. A few incident reports may include some tabular data, such as the time and location, but the tabular data is missing in most incident reports. Therefore, the incident data has a strong temporal and spatial correlation because natural language is sequential, as the meaning of a word depends on the words that precede or follow it. However, traditional machine learning treats data (words) independently distributed in the context by following certain patterns that can be found statistically. Hochreiter and Schmidhuber proposed the first LSTM model [32], which is an advanced form of recurrent neural network (RNN), as it introduces "memory" and "forget" cells. These cells can effectively resolve problems such as vanishing gradient and long-term dependence with which RNNs struggle. This study uses an LSTM neural network model to process word vectors and make classifications.

The overall procedure of our model is shown in Figure 3. As mentioned in Section 1, we approach the problem by developing models that can identify the primary and contributing factors of the ASRS incident reports based on deep recurrent neural networks. Specifically, we start with a general unsupervised language model called Universal Language Model Fine-Tuning (ULMFiT), thoroughly trained by Wikipedia articles [33]. Next, we use an inductive transfer learning technique to refine this general model on our specific ASRS data sets to get familiar with the structure and semantics of the narrative text in the incident reports. Inspired by [34], we implement a universal language model based on Averaged Stochastic Gradient Descent Weight-Dropped LSTM (AWD-LSTM), a state-of-the-art variant of RNNs for language modeling and text classification tasks. The model uses a variety of effective regularization techniques that significantly improve the generalization performance of vanilla LSTM recurrent neural networks. Afterward, using supervised learning and 80% of the incident reports as training data sets, we build and fine-tune classifiers using the AWD-LSTM model and additional concatenation and feed-forward layers to predict primary and multiple contributing factors in the textual reports.

Table 6: The summary of the incident reports and their label distribution in the training set before and after data oversampling, as well as validation and test sets.

	Original	Train (oversampled)	Validation	Test
Human factors (HF)	87356 (62.8%)	87356 (25.4%)	10941 (64.0%)	16145 (63.4%)
Aircraft (AC)	32690 (23.5%)	65380 (19.0%)	3823 (22.4%)	6620 (26.0%)
Company policy (CP)	5335 (3.8%)	53350 (15.5%)	635 (3.7%)	1047 (4.1%)
Procedure (PR)	5321 (3.8%)	53210 (15.4%)	645 (3.7%)	1004 (4.0%)
Weather (WE)	4979 (3.6%)	49790 (14.5%)	623 (3.7%)	952 (3.7%)
Airport (AP)	3424 (2.5%)	34240 (10.0%)	428 (2.4%)	643 (2.5%)
Total	139105 (100%)	343326 (100%)	17095 (100%)	25451 (100%)

Validation and test data are maintained as imbalanced as the original training set to truly represent the data sample distribution.

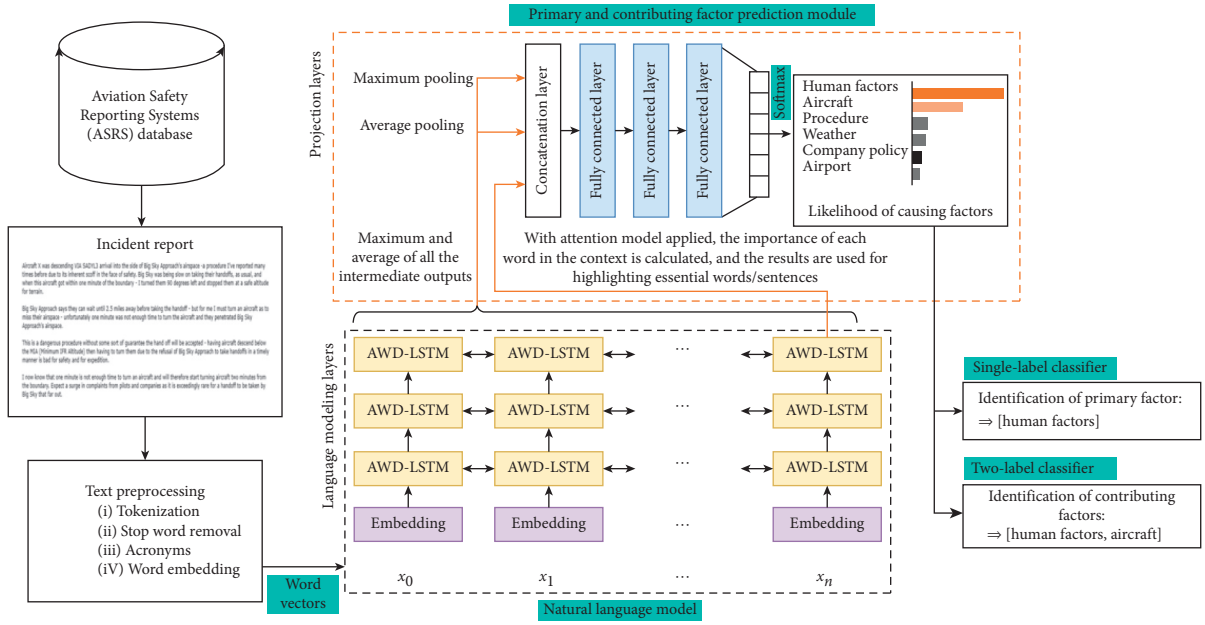

Figure 3: End-to-end diagram of the identification of factors of incident reports. Incident reports including the narratives are downloaded from the ASRS database. After being preprocessed, they are fed to deep neural network model which is composed of two components: (i) a language modeling module, an input layer of embedding and three stacked layers of bidirectional AWD-LSTM recurrent neural networks; and (ii) a prediction module, a flatten layer and three fully connected layers. After processing by these two modules, a probability score is assigned to each factor. Finally, the primary and contributing factors are predicted based on the ranking of probability scores.

We address the identification of the primary factors (single-label) and contributing factors (multilabel) as two different classification tasks, although they share the same architecture until the last layer. It might be tempting to use highest and second-highest probability factors as multilabel results, so that only one model is sufficient to classify multilabel, multiclass tasks. However, the experiment from this study shows inferior results with this approach, as the results are likely to be biased toward dominant factors in the data set. Instead, the training processes for single label and multiple labels have to run separately with corresponding truth labels. Table 3 shows a complete procedure of our approach. After the data preprocessing stage explicitly explained in Section 3, we apply deep neural networks on the textual data. The major steps are explained as follows.

4.2. The Baseline Natural Language Model. Unlike extant research, which does not use any textual data aside from the data used for the primary task of each study and thus restricts the quality and quantity of the data set, we first introduce a universal language model [35] that is pretrained with a large, well-prepared Wikipedia text corpus, thanks to Salesforce Research[2]. The benefits of this approach are threefold: (1) The pretrained open-source model is trained thoroughly. It is called "universal" as it covers a large set of textual data, including most of the words that appear in the incident reports. (2) The amount of available textual data is greatly increased. Even though we have 181,651 incident reports with a total of about 46 million words, this is still not a large enough corpus to train a deep neural network model well. Google[3] recommends a corpus of about 0.8 billion words. (3) This approach saves significant computational resources. Otherwise, a supercomputer would take one month to train a well-prepared language model, which is not feasible for most academic researchers.

4.3. Baseline Language Model Fine-Tuning. We have a well-made baseline natural language model, but the problem is that it seems to be unrelated to our specific task. After all, the

incident narrative data is different from the Wikipedia text corpus. This is where fine-tuning comes into play [36]. To make the baseline language model suited to our specific task, we refine our universal language model using the ASRS data set. Inspired by [34], we implement a universal language model based on AWD-LSTM.

4.4. Prediction of Primary and Contributing Factors. As Figure 3 shows, after the words have been processed by the language model, they are now presented in high-dimensional vectors and fed to artificial neural networks (ANNs) to generate the prediction. Extant research has proven ANNs to be successful at classification tasks [37]. Naturally, the one having the highest probability score among the six factors should be identified as the primary factor. However, due to the imbalance of the sample data and the narrative texts' intrinsic complexity, we apply novel adjustable thresholds to "human factors" only to control the rate of false positives, as discussed in more detail in Section 5. No threshold is applied to other primary factors or when identifying multiple contributing factors. In this way, we achieve a good balance among the six most common primary factors in the overall performance without adding too much complexity.

5. Experimental Setup and Result Discussion

As shown in Table 4, each report contains one primary factor and an average of 1.47 contributing factors. Therefore, we design the model to predict up to two contributing factors for each incident report after weighing the advantages and disadvantages of additional complexity. In this study, two classifiers are developed: (i) a *single-label classifier* to predict the primary factor and (ii) a *multilabel classifier* to predict up to two contributing factors. These two classifiers follow the same methodology explained in Section 4, except that different truth labels and label sets are used during the training step. This is a clear example of the adaptability and reusability of deep learning models. Usually, only the project layers need updates when the task is changed, while the main model remains the same. We will discuss the details of our experimental setup and results later in this section.

5.1. Configuration. In this section, we briefly discuss the configuration and critical hyperparameters of our model, that is, learning rate, batch size, hidden layer size, dropout, and so forth. We use a grid search algorithm [38] to find the optimal values that lead to the highest performance on the training set.

Both primary and contributing identification classifiers use a three-layer LSTM[4] model with 1152 hidden units in the hidden layer. We train our model on a Tesla V100-SXM2 GPU machine with 16 GB of memory. We use a batch size of 128 as optimum, based on the computing stability of the stochastic gradient descent and memory restrictions of the GPU machine. Each word is vectorized to 400 dimensions using a vocabulary size of 60,000. The optimal number of dimensions is often between 300 and 500, according to industry experiments and research [39]. In this study, the maximum length of a sequence is set to 700 words to avoid the diminishing returns of larger networks [40]. As shown in Figure 2, most of the incident reports have no more than 700 words; for reports having more words, all words beyond 700 are simply truncated and ignored. Thus, the input shape is (128, 700, 400).

As mentioned in Section 4, the deep RNN language model is based on the AWD-LSTM, which uses dropouts on the recurrent weights for effective regularization and prevents the model from overfitting. As a means of regularization, such dropouts can effectively reduce the overfitting problem [41]. In this study, the dropout values for the embedding, input/output of every intermediate layer, the output of the final layer, and the hidden-to-hidden weights (recurrent weight-dropped) are 0.25, 0.15, 0.1, and 0.2, respectively.

To train our deep neural network's parameters with ASRS incident reports, we use Slanted Triangular Learning Rate [33]. It quickly increases within the first few hundred iterations and then gradually decays until the epoch ends. This dynamic learning rate enables the model to learn quickly when the loss is high in the beginning and to gradually refine the parameters when the loss becomes smaller[5].

5.2. Retraining Effect on Language Modeling and Factor Identification. As mentioned in Subsection 4.3, AWD-LSTM, initially trained on a well-prepared wiki text corpus, is our baseline LSTM model. It is retrained using the ASRS data set to make it work well in this study. Such retraining is especially useful if the text data of the target task is massive. Figure 4 shows how the training loss, validation loss, and prediction accuracy of the language model change during the training epochs. Each epoch takes about 45 minutes to complete. Initially, the training loss and validation loss are reduced, and the accuracy gradually improves, which indicates that the model can make better predictions in each epoch. In other words, the model is learning. After certain epochs, in our case, after the 15^{th} epoch, training loss continues to decrease linearly, while validation loss and accuracy stabilize at certain values, indicating the optimal time to terminate training; otherwise, the model will overfit on the training set, a notorious problem in deep learning [42]. In our study, retraining the language model improves the identification accuracy of the primary factor by 3.6%, consistent with the retraining gain described in the literature [33, 43].

5.3. Evaluation Metrics. Primary factor identification results are normalized to prevent the results of dominant classes from weighing too much. Therefore, in this study, percentages of true positives, false positives, and false negatives, rather than their counts, are used to calculate the precision and recall. Normalization puts more weight on rare classes, and this is usually more reasonable to measure classes that are not evenly distributed [44].

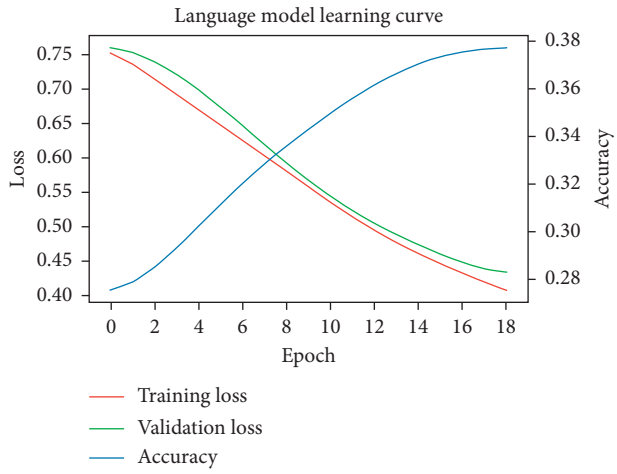

FIGURE 4: Language model learning curve. Accuracy here is defined as the percentage of predicted correct next word from a given vocabulary. Initially, the language model AWD-LSTM can achieve an accuracy of only 0.28; after 15 epochs, the accuracy improved to 0.38, a significant boost.

An "exact match" metric makes sense to evaluate the performance of the primary causal factor identification, as there is only one primary factor for each incident report. However, "exact match" does not work very well for evaluating the performance of multiple causal factor identification, because "exact match" completely ignores partial correctness. Thus, [45] introduces 11 common evaluation metrics for multiple causal factor (multilabel) identification. In this paper, hamming loss, micro-F_1, and macro-F_1 are selected to measure our results, as these three are commonly recognized and chosen in previous research [13, 46].

Hamming loss is the fraction of labels that are incorrectly predicted. Unlike "exact match," hamming loss is more forgiving in that it penalizes only the individual labels that do not match the truth labels [47]. Hamming loss is a loss function; thus the lower, the better.

Besides the hamming-loss metric, macro-F_1 and micro-F_1 are two conventional methods to evaluate the performance of multiple causal factor identification [48]. The critical distinction between macro-F_1 and micro-F_1 is that macro is an average per category, while micro is an average per sample point. These metrics are computed according to the following equations:

$$\text{Hamming loss} = -\frac{1}{ml}\sum_{i=1}^{m}\sum_{j=1}^{l}\left[h_{ij} \neq y_{ij}\right], \tag{1}$$

$$\text{macro } F_1 = \frac{1}{l}\sum_{j=1}^{l}\frac{2\sum_{i=1}^{m} y_{ij}h_{ij}}{\sum_{i=1}^{m} y_{ij} + \sum_{i=1}^{m} h_{ij}}, \tag{2}$$

$$\text{micro } F_1 = \frac{2\sum_{j=1}^{l}\sum_{i=1}^{m} y_{ij}h_{ij}}{\sum_{j=1}^{l}\sum_{i=1}^{m} y_{ij} + \sum_{j=1}^{l}\sum_{i=1}^{m} h_{ij}}, \tag{3}$$

where h_{ij} is the target, y_{ij} is the prediction, m is the number of samples, and l is the number of labels.

5.4. Primary Factor (Single-Label) Identification Performance. As "human factors" still account for 25.4% of all incidents after oversampling, the classifier tends to be biased toward "human factors." To further reduce the bias, we apply a confidence threshold to control the percentage of false positives in the "human factors" category. For example, a confidence threshold equal to 0.55 means that the classifier only labels an incident with "human factors" if it has 55% confidence or more; otherwise, the category with the second-highest confidence, even it is lower than HF, is chosen. See Table 7 for an example.

Primary factor identification results are shown in Table 8. We apply the threshold to the "human factors" class only to reduce the rate of its false positives because it greatly outnumbers the other classes. Based on our experiments with different thresholds starting from 0.3 to 0.7 with increment of 0.05, we find that an HF threshold of 0.55 effectively reduces the rate of HF's false positives. Considering that the data samples of each factor are imbalanced, we believe that micro-F_1 is a better way to assess the model's performance because micro-F_1 is an average per sample point (see equation (3)). As shown in Table 9, the micro-F_1 scores of all classes except WE are improved (Tables 8 and 9).

5.5. Contributing Factors (Multilabel) Identification Performance. In this study, each incident's contributing factors are prepared by combining the original primary and contributing factors (if any) of the incidents. An example is shown in Table 10.

As mentioned in Section 5, our model is designed to predict up to two factors for each incident report. Consequently, any prediction is definitely a mismatch for incidents that are labeled with more than two factors. Nevertheless, multilabel evaluation metrics consider partial match (see equations (1)-(3) in Section 5.3). Table 11 summarizes the multilabel performance of our model by each category and overall performance. Our model achieves an F_1 score of 0.763 by averaging four averages: micro-avg, macro-avg, weighted-avg, and sample-avg. As shown in Table 5, "human factors" and "aircraft" significantly outnumber the other four categories combined. Therefore, micro-avg, calculated by counting true positives, false negatives, and positives globally, is preferable for evaluating our model's performance. Sample-avg, average based on samples, and weighted-avg, average based on labels, are adjusted versions of micro-avg and output similar results. On the other hand, the macro-avg metric can be expected to generate the worst F_1 score as it treats all classes equally, totally ignoring the number of samples in each class. Thus, it is less accurate than the other three metrics due to data imbalance (Table 11).

5.6. Comparison of Our Results to Previous Studies. To better understand our model's performance, we compare our results with previous studies addressing similar tasks, as well as with a base model without fine-tuning. To make the comparison valid and convincing, we use the same data sets as the previous studies. Because single-label and multilabel tasks have different evaluation metrics, we compare them separately.

Table 7: An example of how the "HF threshold" affects the identification result.

	Probability of each factor						
HF threshold	HF	AC	CP	PR	WE	AP	Identification
Threshold = 0	0.42	0.37	0.09	0.03	0.03	0.06	HF
Threshold = 0.55	0.42	0.37	0.09	0.03	0.03	0.06	AC

If an HF threshold is specified, HF will only be identified when its probability exceeds the specified value; otherwise, the factor with the second-highest confidence is chosen. In this way, the bias toward the dominant factors is well compensated by tuning the threshold. Threshold = 0 (no threshold). Threshold = 0.55.

Table 8: Comparison of the confusion matrix of the single label with and without the threshold (orthogonal values highlighted).

		Predicted label					
Threshold = 0							
		HF	AC	PR	WE	CP	AP
Truth label	HF	0.92	0.05	0.01	0.01	0.01	0.01
	AC	0.16	0.82	0	0.01	0.01	0
	PR	0.57	0.05	0.33	0.02	0.03	0
	WE	0.33	0.07	0.01	0.59	0.01	0
	CP	0.51	0.13	0.03	0.02	0.31	0
	AP	0.51	0.07	0	0.02	0.04	0.35
Threshold = 0.55							
Truth label	HF	0.84	0.07	0.03	0.02	0.02	0.02
	AC	0.08	0.89	0.01	0.01	0.01	0.01
	PR	0.37	0.09	0.47	0.02	0.03	0.02
	WE	0.30	0.05	0.03	0.59	0.01	0.02
	CP	0.32	0.16	0.04	0.02	0.42	0.04
	AP	0.34	0.08	0.02	0.03	0.05	0.47

By applying a proper threshold, the model's ability to identify other rarer classes is significantly improved, and overall performance of HF is improved as well.

Table 9: After applying the threshold, the model's overall performance in terms of micro-F_1 score is improved, especially for rarer factors, as precision and recall become more balanced.

	Probability threshold = 0			Probability threshold = 0.55			F_1 score improvement
	Precision	Recall	Micro-F_1	Precision	Recall	Micro-F_1	Percentage
HF	0.306	0.92	0.502	0.373	0.84	0.516	+2.7%
AC	0.689	0.82	0.748	0.664	0.89	0.761	+1.7%
CP	0.756	0.31	0.440	0.778	0.42	0.545	+23.9%
PR	0.868	0.33	0.478	0.783	0.47	0.588	+23.0%
WE	0.882	0.59	0.706	0.855	0.59	0.702	− 0.5%
AP	0.971	0.35	0.514	0.83	0.47	0.596	+16.0%

Table 10: An example of how multiple labels are prepared for each incident report using one-hot encoding.

HF	AC	PR	WE	CP	AP	Truth label
✓	—	—	—	✓	—	[1 0 0 0 1 0]

1 indicates that a factor is present, and 0 indicates that a factor is absent. Matches and mismatches of multiple labels prepared in this way can be conveniently evaluated by the Python scikit-learn library [49].

Table 12 clearly shows that our model is superior to Shi et al.'s [4] in terms of label categories and model accuracy. We not only identify the six most common causal factors but also expand our model to address multiple causal factors. In addition, our HF accuracy is significantly better, while AC accuracy is equivalent. With the improved HF accuracy, the overall accuracy is improved significantly, as it is the most frequent class. Robinson's research [13] is the most closely related study we can find in terms of multilabel classification. He implements a latent semantic analysis algorithm to classify all 16 classes for only 4,497 incident reports, compared with our 138,392 reports for training. As mentioned in Section 1, the ten rarest classes account for less than 5% of total incident reports. Therefore, his research attempts to classify 16 classes with such little data are not very reasonable, and the result is inferior to ours. In addition, the advantages of the fine-tuned language model are also demonstrated, because it refines the word embeddings with the target data set. Table 12 indicates that the LSTM with the fine-tuned language model outperforms the one without fine-tuning by 3.3% on HF accuracy and 1.9% on AC accuracy in single-label classification. In multilabel

Table 11: A summary of our model's performance in identification of multiple causal factors.

	Precision	Recall	F_1 score
HF	0.88	0.93	0.90
AC	0.87	0.83	0.85
PR	0.70	0.46	0.56
WE	0.71	0.43	0.53
CP	0.65	0.37	0.47
AP	0.68	0.39	0.50
Micro-avg	0.84	0.77	0.80
Macro-avg	0.75	0.57	0.63
Weighted-avg	0.82	0.77	0.79
Sample-avg	0.88	0.84	0.83
Hamming loss = 0.091			

Table 12: A performance comparison of our method with previous research, regarding single-label and multilabel identification.

Studies	Algorithm	HF accuracy	AC accuracy	Remark
Shi et al. [4]	Naive Bayes	73.2%	81.1%	This study targets *HF* and *AC* only
	Hoeffding tree	74.9%	87.0%	
	OzaBagADWIN	76.5%	88.3%	
Our study	LSTM without fine-tuned language model	84.8%	85.1%	Our study achieves a better result regarding *HF* and can identify four more factors
	LSTM with fine-tuned language model	88.1%	87.0%	
Studies	Algorithm	Hamming loss	F_1 score	Remark
Robinson [13]	Latent semantic analysis	0.269	0.409	Impractically targeting 16 factors
Our study	LSTM without fine-tuned language model	0.135	0.628	Our study feasibly targets the six most frequent factors with promising results achieved
	LSTM with the fine-tuned language model	0.091	0.763	

The advantage of the deep learning methods over traditional machine learning methods is clearly shown.

classification, the LSTM with the fine-tuned language model has a lower hamming loss but higher F_1 score compared with the base model. To sum up, these results demonstrate that the use of a fine-tuned language model can improve classification accuracy.

6. Implications

We build two classifiers to identify the primary and contributing factors, using a deep recurrent network algorithm. These models are trained with the narrative texts of ASRS incident reports. With our classification models, the amount of incident report analysis done by human experts can be significantly reduced. When an incident report is generated, our first classifier identifies the primary factor and then properly indexes it into the database. Then, the second classifier identifies additional contributing factors. Our model can automate most of the tasks, and human experts may only need to check the incidents classified with low confidence by our model. The implications of our study are summarized in four perspectives presented below.

First, from the perspective of aviation safety reviewers, our study can help them facilitate the identification of causal factors. As demonstrated in Section 5, our model achieves an average accuracy of 82% on the six most common factors and about 89% on the two most common factors on average. In addition, our model has achieved the best multilabel, multiclass identification results compared with extant research. Our study has shown that this approach can identify causal factors for 95% of incident reports in the database with little human intervention. If they adopt our approach, aviation incident reporting systems can quickly issue initial results to relevant parties, such as air traffic controllers, airline companies, and airport authorities.

Second, incident reports that are identified with high confidence by our models do not require review by safety experts. Less than 4.7% of incident reports are predicted with low confidence (probability threshold $\leq$ 0.55). Safety experts may only need to review those incident reports to make sure causal factors are correctly identified. Figure 5 is an example of an incident report parsed by our model with an attention mechanism applied [50]. The attention mechanism is an algorithm to calculate each word and sentence's relative importance based on the required outputs. For instance, if the truth label (the output) is "aircraft," then words and sentences likely to be related to "aircraft" are assigned higher importance or probability in the incident texts. As Figure 5 shows, the highlighted words and sentences are likely the critical information associated with the true causal factors of the incident. These highlights can help safety experts locate

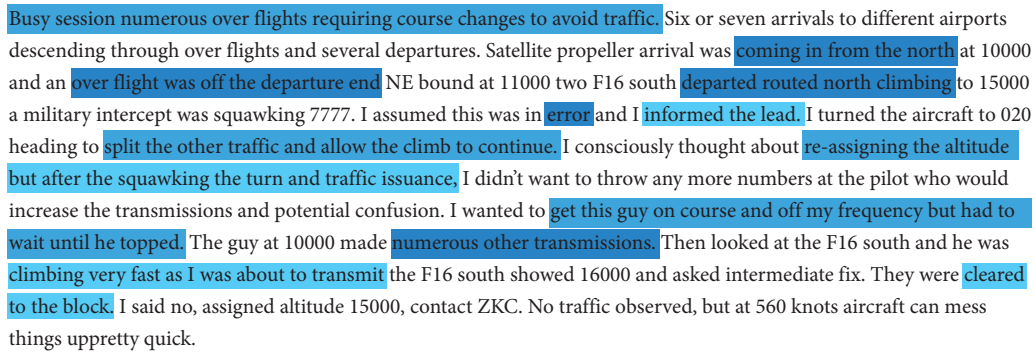

Figure 5: Example of narrative texts that have been processed by our model. With the attention mechanism application, the potential essential sentences or words are highlighted, and they are more comfortable for human experts to review.

the definitive information faster, which substantially expedites the manual review process. At the same time, safety experts' correct labeling of manually reviewed incident reports can improve the model's performance in the long run. This model can further evolve into a text summarization system by generating a "Synopsis" [51], which currently has to be generated by safety experts manually. By reviewing the "Synopsis" generated from each incident report, the number of incidents that a human expert can handle per unit of time is greatly increased.

Third, from the perspective of reporting systems, such automation makes the generation of statistical reports easier. Due to the voluntary nature of the reports submitted to ASRS, NASA mainly uses the data as a lower-bound estimate. For example, there were 112,305 human error incident reports submitted to the ASRS from January 1988 through July 2020. It can be confidently concluded that at least 112,305 human errors contributed to aviation incidents during this period. Based on this lower-bound estimate, decision-makers can determine whether a problem exists and requires further investigation [52]. It is easy to provide aggregated and even dynamic incident statistics once the causal factor identification is automated with satisfactory performance.

Fourth, the deep learning solution developed in this study, a very versatile technique, can be redesigned and adapted to different domains other than aviation. This study has chosen the ASRS as an explicit example to show how deep learning techniques can help safety experts process a large quantity of textual data quickly and accurately. The application of this technique can help aviation safety experts find emerging dangers and potential hazards promptly from a large volume of incident reports. Although the incident reports in other transportation domains might be different in terms of quantities, textual characteristics, report formats, and so forth, the methodology designed in this paper can be adapted to address those varied tasks.

7. Conclusion and Limitations

Incident report analysis is crucial to improve safety management in high-risk work environments. Though a large amount of incident data is generated every day with the advances in data storage management and Internet of Things (IoT), effective and timely utilization of these resources has been hampered by the tremendous human effort needed to identify incident causes. This study presents models that can automate causal factor identification of ASRS incident reports based on deep recurrent neural networks. Our results demonstrate that deep recurrent neural network algorithms, trained and fine-tuned with proper transfer learning techniques, are versatile enough to build classifiers to predict the primary factor or multiple factors with minor modifications. Therefore, an initial understanding of incident reports' factors can be gained from automated incident report analysis. Given these potential benefits, this study's promising results may encourage researchers to explore the application of deep learning algorithms to other domains, such as autotransportation, medical facilities, information technology failure, and injury reporting, where automated text analysis is much needed.

There are several limitations to this deep learning approach. Currently, we are only able to classify the six most frequent categories in ASRS data sets. Ten other much rarer categories, accounting for approximately 5% of all incident reports, are unaddressed, primarily due to the lack of sufficient sample data for training the deep learning approach. Additional efforts will be required to find a deep learning architecture that requires less data or to figure out effective ways to augment the limited data samples. Another limitation of our study is that we have limited our multilabel classifier to no more than two factors. However, about 9% of incident reports have more than two labels. A more sophisticated model may further improve identification accuracy. Finally, tabular data such as locations and time periods are not used in the deep learning model proposed in this study. Future studies can investigate the causal relationships between tabular data and incident factors to determine which locations or time periods are more likely to be associated with human factor-related incidents.

Data Availability

The data used in this paper was collected from asrs.arc.nasa.gov/search/database.html. Researchers can request the data from the ASRS, or they can download it from the website.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by Trinity University's Faculty Research Start-up Fund and Summer Research Stipend Program for 2018.

References

[1] T. Ali, H. Khazaei, M. H. Y. Moghaddam, and Y. Hassan, *Machine Learning in Transportation*, Hindawi, London, UK, 2019.

[2] L. Tanguy, N. Tulechki, A. Urieli, E. Hermann, and C. Raynal, "Natural language processing for aviation safety reports: from classification to interactive analysis," *Computers in Industry*, vol. 78, pp. 80–95, 2016.

[3] D. Harris and W.-C. Li, *Decision Making in Aviation*, Routledge, Oxfordshire, UK, 2017.

[4] D. Shi, J. Guan, J. Zurada, and A. Manikas, "A data-mining approach to identification of risk factors in safety management systems," *Journal of Management Information Systems*, vol. 34, no. 4, pp. 1054–1081, 2017.

[5] G. D. Edkins, "The indicate safety program: evaluation of a method to proactively improve airline safety performance," *Safety Science*, vol. 30, no. 3, pp. 275–295, 1998.

[6] C. Posse, B. Matzke, C. Anderson, A. Brothers, M. Matzke, and T. Ferryman, "Extracting information from narratives: an application to aviation safety reports," in *Proceedings of the 2005 IEEE Aerospace Conference*, pp. 3678–3690, Big sky, MT, USA, March 2005.

[7] R. S. Dodd, D. Eldredge, and S. J. Mangold, *A Review and Discussion of Flight Management System Incidents Reported to the Aviation Safety Reporting System*, The National Academies of Sciences, Engineering, and Medicine, Washington, DC, USA, 1992.

[8] J. Orasanu and L. Martin, "Errors in aviation decision making: a factor in accidents and incidents," in *Proceedings of the workshop on human error, safety, and systems development*, pp. 100–107, Citeseer, Glasgow, Scotland, 1998.

[9] M. Bao and S. Ding, "Individual-related factors and management-related factors in aviation maintenance," *Procedia Engineering*, vol. 80, pp. 293–302, 2014.

[10] T. Edwards, S. Sharples, J. R. Wilson, and B. Kirwan, "Factor interaction influences on human performance in air traffic control: the need for a multifactorial model," *Work*, vol. 41, pp. 159–166, 2012.

[11] J.-P. Antoine, B. Rajagopalan, and D. Bowman, "Automated content analysis for construction safety: a natural language processing system to extract precursors and outcomes from unstructured injury reports," *Automation in Construction*, vol. 62, pp. 45–56, 2016.

[12] S. Mousa, S. Soleimani, J. Codjoe, and M. Leitner, "A comprehensive railroad-highway grade crossing consolidation model: a machine learning approach," *Accident; Analysis and Prevention*, vol. 128, pp. 65–77, 2019.

[13] S. Robinson, "Multi-label classification of contributing causal factors in self-reported safety narratives," *Safety*, vol. 4, no. 3, p. 30, 2018.

[14] L. N. Tiller and J. P. Bliss, "Categorization of near-collision close calls reported to the aviation safety reporting system," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 61, no. 1, pp. 1866–1870, 2017.

[15] K. D. Kuhn, "Using structural topic modeling to identify latent topics and trends in aviation incident reports," *Transportation Research Part C: Emerging Technologies*, vol. 87, pp. 105–122, 2018.

[16] M. Abedin, V. Ng, and L. Khan, "Cause identification from aviation safety incident reports via weakly supervised semantic lexicon construction," *Journal of Artificial Intelligence Research*, vol. 38, no. 1, pp. 569–631, 2010.

[17] C. Dong, C. Shao, J. Li, and Z. Xiong, "An improved deep learning model for traffic crash prediction," *Journal of Advanced Transportation*, vol. 2018, Article ID 3869106, 13 pages, 2018.

[18] B. Cortez, B. Carrera, Y.-J. Kim, and J.-Y. Jung, "An architecture for emergency event prediction using lstm recurrent neural networks," *Expert Systems with Applications*, vol. 97, pp. 315–324, 2018.

[19] X. Zhang and S. Mahadevan, "Bayesian neural networks for flight trajectory prediction and safety assessment," *Decision Support Systems*, Article ID 113246, 2020.

[20] X. Luo, D. Li, Yu Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with knn and lstm," *Journal of Advanced Transportation*, vol. 2019, Article ID 4145353, 10 pages, 2019.

[21] C. D. Manning, "Probabilistic syntax," in *Probabilistic Linguistics*, MIT Press, Cambridge, MA, USA, 2003.

[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, pp. 3111–3119, Curran Associates Inc., New York, NY, USA, 2013.

[23] M. M. Najafabadi, F. O Villanustre, and T. M. Khoshgoftaar, "Naeem Seliya, and Randall Wald. Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, 2015.

[24] S. T Aroyehun and A. Gelbukh, "Aggression detection in social media: using deep neural networks, data augmentation, and pseudo labeling," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, Santa Fe, NM, USA, 2018.

[25] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of the International Conference on Learning Representations*, Scottsdale, AZ, USA, 2013.

[26] F. C. Bernardini, B. Rodrigo da Silva, M. Rodrigo, and E. B. Mitacc Meza, "Cardinality and density measures and their influence to multi-label learning methods," *Learning and Nonlinear Models*, vol. 12, no. 1, pp. 53–71, 2014.

[27] A. Kosmopoulos, I. Partalas, E. Gaussier, G. Paliouras, and I. Androutsopoulos, "Evaluation measures for hierarchical classification: a unified view and novel approaches," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 820–865, 2015.

[28] J. Nothman, H. Qin, and Y. Roman, "Stop word lists in free open-source software packages," in *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, Melbourne, Australia, 2018.

[29] E. Loper and H. Steven Bird, "Nltk: the natural language toolkit," in *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, Association for Computational Linguistics, Philadelphia, PA, USA, 2002.

[30] N. Mahmoudi, P. Docherty, and P. Moscato, "Deep neural networks understand investors better," *Decision Support Systems*, vol. 112, pp. 23–34, 2018.
[31] H. Moradi, W. Wang, A. Fernandez, and D. Zhu, "Upredict: a user-level profiler-based predictive framework for single Vm applications in multi-tenant clouds," 2019, http://arxiv.org/abs/1908.04491.
[32] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[33] J. Howard and S. Ruder, "Fine-tuned language models for text classification," 2018, https://arxiv.org/abs/1801.06146.
[34] S. Merity and R. Socher, "Regularizing and optimizing LSTM language models," 2017, https://arxiv.org/abs/1708.02182.
[35] E. Grave, J. Armand, and N. Usunier, "Improving neural language models with a continuous cache," 2016, https://arxiv.org/abs/1612.04426.
[36] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.
[37] C. Gleue, D. Eilers, H.-J. von Mettenheim, M. H. Breitner, and Breitner, "Decision support for the automotive industry," *Business & Information Systems Engineering*, vol. 61, no. 4, pp. 385–397, 2019.
[38] J. Y. Hesterman, L. Caucci, M. A. Kupinski, H. H. Barrett, and L. R. Furenlid, "Maximum-likelihood estimation with a contracting-grid search algorithm," *IEEE Transactions on Nuclear Science*, vol. 57, no. 3, pp. 1077–1084, 2010.
[39] Zi Yin and Y. Shen, "On the dimensionality of word embedding," in *Advances in Neural Information Processing Systems*Curran Associates Inc., New York, NY, USA, 2018.
[40] K. Greff, R. Srivastava, K. Jan, and B. R. Steunebrink, "Lstm: a search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2016.
[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, 2014.
[42] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, Cambridge, UK, 2016, http://www.deeplearningbook.org.
[43] M. E. Peters, M. Neumann, M. Iyyer et al., "Deep contextualized word representations," 2018, https://arxiv.org/abs/1802.05365.
[44] B. Peter, "The normalized recall and related measures," *SIGIR Forum*, vol. 17, no. 4, pp. 122–128, 1983.
[45] X. Zhu, "A unified view of multi-label performance measures," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 3780–3788, Sydney, Australia, 2017.
[46] P. Probst, Q. Au, G. Casalicchio, C. Stachl, and B. Bischl, "Multilabel classification with R package mlr," *The R Journal*, vol. 9, no. 1, pp. 352–369, 2017.
[47] G. Tsoumakas and I. Katakis, "Multi-label classification," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.
[48] Y. Yang, "An evaluation of statistical approaches to text categorization," *Information Retrieval*, vol. 1, no. 1-2, pp. 69–90, 1999.
[49] F. Pedregosa, G. Varoquaux, A. Gramfort et al., "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[50] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of the International Conference on Learning Representations*, San Diego, CA, USA, 2016.
[51] M. Moradi, G. Dorffner, and M. Samwald, "Deep contextualized embeddings for quantifying the informative content in biomedical text summarization," *Computer Methods and Programs in Biomedicine*, vol. 184, Article ID 105117, 2020.
[52] "ASRS database statistics," 1994, https://asrs.arc.nasa.gov/publications/directline/dl8_stat.htm.