

Hierarchical Tree-based Sequential Event Prediction with Application in the Aviation Accident Report

Xinyu Zhao^{*†}, Hao Yan^{*†}, Yongming Liu^{†‡},

^{*}School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281, USA

[†]School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ, 85287, USA

[‡]{xinyu.zhao,, HaoYan,yongming.liu}@asu.edu

Abstract—Sequential event prediction is a well-studied area and has been widely used in proactive management, recommender systems and healthcare. One major assumption of the existing sequential event prediction methods is that similar event sequence patterns in the historical record will repeat themselves, enabling us to predict future events. However, in reality, the assumption becomes less convincing when we are trying to predict rare or unique sequences. Furthermore, the representation of the event may be complex with hierarchical structures. In this paper, we aim to solve this issue by taking advantage of the multi-level or hierarchical representation of these rare events. We proposed to build a sequential Encoder-Decoder framework to predict the event sequences. More specifically, in the encoding layer, we built a hierarchical embedding representation for the events. In the decoding layer, we first predict the high-level events and the low-level events are generated according to a hierarchical graphical structure. We propose to link the encoding decoding layers with the temporal models for future event prediction. In this article, we further discussed applying the proposed model into the failure event prediction according to the aviation accident reports and have shown improved accuracy and model interpretability.

Index Terms—event prediction, aviation accident, hierarchical tree structure, embedding, sequential model

I. INTRODUCTION

Sequential event prediction or sequential pattern mining is a well-studied topic in the literature. There are a lot of real-world scenarios where the data is released sequentially. People believe that there exist repetitive patterns of event sequences so that we can predict future events. For example, many companies build their recommender systems to predict the next possible product for the users according to their purchase history. The healthcare system discovers the relationships among patients' sequential symptoms to mitigate the adverse effect of a treatment (drugs or surgery). Modern engineering systems like aviation/distributed computing/energy systems diagnosed failure event logs and took prompt actions to avoid disaster when a similar failure pattern occurs.

Most of the applications mentioned above can only make predictions with simple flat representation. When it comes to the rare event sequence with a more complex hierarchical representation, it may lose the prediction power. However, these complex hierarchical representation is very common in the following applications. 1) In the recommender system, we always see that the online store is organized as *category* \rightarrow *subcategory*. Such a hierarchical structure is frequently applied in recommendation algorithms. It has been shown that

the recommendation performance would be improved through incorporating the hierarchical structure since the similarity in the hierarchical layer means similar properties in items and similar preferences in users [1]. 2) In the healthcare system, accurate prediction of patients' symptoms makes personalized healthcare possible, and it relies on the large volume of the medical record. However, most patients only have a limited number of records [2], and some symptoms are quite rare to find. To solve this, the patients and symptoms can be grouped together for better prediction for patients with rare symptoms. 3) In the engineering systems, many maintenance or event logs exist to describe the repetitive failure event sequence to take prompt actions [3]. However, failure events tend to be unique and rare, which makes them hard to predict. Similarly, to tackle the sparsity issue of the failure events, a proper hierarchical structure can be built to describe the failure event so that the failure event can be modeled at different granular levels. In this work, we focus on building a scalable algorithm for the hierarchical event prediction problem. Our algorithm is trying to deal with the problem from three main directions:

- Since the number of the possible events is very large, representing the event using one-hot-encoding will lead to a high dimensional vector space. Thus, we represent the event using hierarchical encoder and decoder embedding layers for the multi-level events. The result further shows that the embedding representation is beneficial for the data sparsity issue since similar events tend to close to each other in the embedding space.
- Our proposed models combine the hierarchical encoder/decoder layers with the sequential dynamic models such as the Recurrent Neural Network to capture long-term dependencies between events.
- The proposed algorithm is applied to analyze the aviation accident report. We have shown both great interpretability of the learned embedding coefficients and better event prediction accuracy.

In this article, we will bring our algorithm into a novel application for the proactive management of the aviation system. The aviation event log data is collected by the National Transportation Safety Board (NTSB) [4]. From Table. I, we can see the hierarchical structure of the accident events. "*Cruise*" denote the phase of flight. "*Rotor failure/malfunction*" further describes what happened during the cruise phase. "*Rotor drive*

system, tail rotor drive shaft” introduces more details about the problem with the rotor. Accurate modeling of the event sequences is important to quantify the risk of aircraft [5].

Phase	Occurrence	Subject
Cruise	Rotor failure/malfunction	Rotor drive system, tail rotor drive shaft
Cruise	Rotor failure/malfunction	Lubricant, grease
Descent - emergency	Forced landing	Autorotation
Emergency landing	Hard landing	Terrain condition

TABLE I: An example of the event log from NTSB. Each event can be described using three levels of information. *Phase* means the flight phase of the aircraft. *Occurrence* denotes the key indicators for the accident. *Subject* denotes the detailed description of the occurrence.

Our experiment shows that the proposed method outperformed the traditional methods, which can benefit a lot from the hierarchical representation of failure events with improved interpretability. This paper is organized as follows. In section 2, we review the traditional method for sequential event prediction, hierarchical event prediction, and the applications in aviation accident report analysis. In Section 3, we describe the proposed Hierarchical Tree-based Sequential Event Prediction model. We further show a case study with the NTSB aviation accident data in Section 4. Finally, we conclude the paper in Section 5.

II. RELATED WORK

In this section, we will review related works in the literature. In particular, we will focus on three different areas, event sequence prediction, hierarchical event prediction, and their applications in the aviation accident report analysis.

A. Event Sequence Prediction

Here, we will review the literature related to the event sequence problem. For example, the association-rule based methods have been proposed to predict the next event over time [6]. Network-based method [7] is used to represent the events pairs as edges and predict the events based on temporal link prediction algorithm and node ranking. The recurrent neural network (RNN) is used to combine the event order and the duration of the event to predict the next event data [8]. Another RNN-based work design a novel component modulator for appending the attributes to the event [9]. However, all aforementioned methods are not designed for the complex hierarchical structures of the events, which is common in many engineering systems. The literature related to incorporate more complex hierarchical data structure into the sequential prediction framework will be discussed in the next section.

B. Hierarchical Event Prediction

Due to the flexibility of the neural network architecture, several deep-learning-based frameworks are developed to learn the hierarchical embedding for the events and then predict the

future event based on the embedding representation. Hu et al. proposed a context-aware hierarchical Long Short Term Memory (LSTM) for the event prediction. The model encodes different levels of events using a two-level LSTM layer [10]. Different levels of the products have been incorporated into the personalized product search [11]. Overall, the deep-learning-based frameworks have been shown to be powerful in capturing complex sequential relationships. In this work, we further show that we can improve the performance of the aforementioned frameworks by adding the hierarchical tree structure on the decoding layer.

III. METHODOLOGY

In this section, we will develop the methodology to model the aviation event sequences. We will start with the problem formulation and taxonomy for the aviation events in Section III-A. The entire proposed framework is then discussed in Section III-B.

A. Hierarchical Taxonomy for Aviation Events and Problem Formulation

We first would like to define our problem mathematically. Given the hierarchical structure of the event representation, we propose to represent each event as a tuple $e^t = (p^t, o^t, s^t)$, which each corpus refers to the Phase, Occurrence, Subject. Some important phases of the aircraft include *take-off*, *cruise*, *descent*, *landing*, etc. Given each phase, there may be different types of event occurrences, including the *loss-of-control*, *collision*, etc. Finally, the subject-level information is a more detailed description of the occurrence, such as the different collision types. However, given that p^t, o^t, s^t are the one-hot encoding representation of the events, the representation of the aviation event may be high-dimensional and inefficient.

Furthermore, the proposed algorithm can be also generalized to events with more layers of the hierarchy, such as $(e_1^t, e_2^t, \dots, e_h^t)$, where the e_1^t is the top-level event category, e_i^t is the i^{th} -level event category. The goal is to predict the next event e^{t+1} given the entire event history $\{e^t\}_{t=1, \dots, T}$.

Finally, the goal of the proposed framework is to predict the next event e^{t+1} given the entire history of event $\{e^{t'}; t' = 1, \dots, t\}$ considering the event hierarchical representation $e^t = (e_1^t, e_2^t, \dots, e_h^t)$.

B. The Proposed Event Prediction with Hierarchical Encoder Decoder

In this section, we will discuss our method for event prediction. Moreover, we aim to learn the embedding representation of the failure event sequence consisting these three level of hierarchy. Here the proposed method would consist three components:

- 1) **Hierarchical encoder network** $P(z_t|e^t; \theta_e)$ with encoder parameter θ_e , which focuses on compressing the high-dimensional event representation e_t into a lower-dimensional feature vector z_t .
- 2) **Sequential transition model** $h_{t+1}, c_{t+1} = f(c_t, z_t; \theta_{tr})$ with transition parameter θ_{tr} , which will be used to

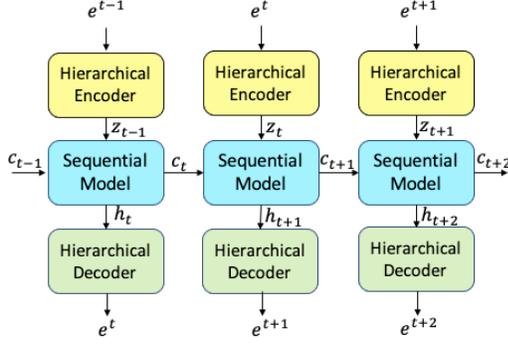


Fig. 1: Model Architecture

model the dynamic transition of such latent lower-dimensional feature vector over time

- 3) **Hierarchical tree-based decoder** $P(e^{t+1}|h_{t+1}; \theta_d)$ with decoder parameter θ_d , which use the tree structure of the events hierarchy to predict the future event e^{t+1} given the feature vector h_{t+1} .

Fig. 1 shows the overall architecture that combines the hierarchical encoder, sequential transition model, and hierarchical tree-based decoder and how they can work together to predict the future event e^{t+1} from the event history $\{e^{t'}\}_{t' < t}$.

Finally, these three components will be trained in an end-to-end manner by maximizing the overall likelihood function $P(\{e^t\}_{t=1}^T; \theta)$, where $\theta = \{\theta_e, \theta_{tr}, \theta_d\}$ are all the model parameters. By decoupling the joint likelihood $P(\{e^t\}_{t=1}^T; \theta) = P(e^1; \theta)P(e^2|e^1; \theta) \cdots P(e^T|\{e^t\}_{t=1}^{T-1}; \theta) = \prod_t P(e^t|\{e^{t'}\}_{t'=1}^t; \theta)$ and assume that h_t represented the information collected through sequence $\{e^{t'}\}_{t'=1}^t$, we can represent the decoder only based on the vector h_t as $P(e^t|\{e^{t'}\}_{t'=1}^t; \theta) = P(e^t|h_t; \theta_d)$.

In the following, we will discuss the mathematical details of the hierarchical encoder network, sequential model, and the hierarchical tree-based decoder in detail.

1) *Hierarchical Encoder* : For event at time t , we first project (p^t, o^t, s^t) to their corresponding embedding layer parameters $U \in \mathbb{R}^{n_c \times |P|}$, $V \in \mathbb{R}^{n_c \times |O|}$, $W \in \mathbb{R}^{n_c \times |S|}$ as $\tilde{p}_t = \sigma(U p^t)$, $\tilde{o}_t = \sigma(V o^t)$, $\tilde{s}_t = \sigma(W s^t)$, where $|P|$ is the number of phases, $|O|$ is the total number of occurrences, $|S|$ is the total number of subject code, and n_c is the size of the embeddings. Furthermore, these three embedding are concatenated into $v_t = [\tilde{p}_t \ \tilde{o}_t \ \tilde{s}_t]$. We further add a linear layer to consider the complex correlation between and within different levels. An activation layer is added at the end to incorporate the non-linearity with $z_t = \sigma(H v_t)$. Here, we can use the ReLU layer for the activation function $\sigma(\cdot)$. Here, the overall framework of the hierarchical encoder is given in Fig. 2.

2) *Sequential Model*: To model the transition of event history, we will use consider the following two different sequential models, the Markovian model and the LSTM model.

- **Markovian Model**: Here, we assume that the future event e^{t+1} only depends on the encoded feature z_t , where

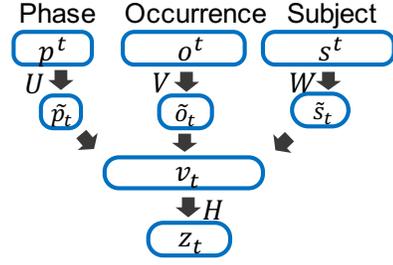


Fig. 2: Hierarchical Encoder

$P(e^{t+1}|z_t; \theta_{tr})$. Moreover, we will model the events such as the phase, occurrence, and subject given z_t separately.

- **Long Short Term Memory**: LSTM is a kind of Recurrent Neural Network which is also able to capture the temporal dependency through a recurrent architecture. Comparing to the standard RNN, LSTM is able to deal with long term dependency. In general, LSTM deals with the long term dependency by adding a cell state and the gating technique. The cell state keeps all the important information from past events, and the gate decides which information is important.

$$c_{t+1}, h_{t+1} = f_{lstm}(c_t, z_t; \theta_{tr}),$$

where f_{lstm} is the standard LSTM model. c_t is the combination of the cell state and memory state in the LSTM model. h_t is the output of the LSTM model at time t and will be used in the decoder.

3) *Hierarchical Tree-based Decoder* : After the hidden state h_t is obtained from previous events, we need to devise a decoding function to predict the failure event in next stage. Notice that our failure event is composed of phase, occurrence and subject according to the taxonomy. Thus, we propose to decouple the joint probability using the hierarchical tree structure by assuming each layer only depends on the previous layer as $P(p_i^{t+1}, o_j^{t+1}, s_k^{t+1}|h_t; \theta_d) = P(p_i^{t+1}|h_t; \theta_d)P(o_j^{t+1}|p_i^{t+1}, h_t; \theta_d)P(s_k^{t+1}|o_j^{t+1}, h_t; \theta_d)$ and the probability of each term can be calculated through the Softmax function. Here, $\theta_d = \{U', \{V'_i\}_{i=1, \dots, |P|}, \{W'_j\}_{j=1, \dots, |O|}\}$ is the decoder parameter (i.e., also known as the decoder embedding parameter), which includes the decoder parameters for phase $U' \in \mathbb{R}^{n' \times n_p}$, decoder parameters for occurrence under phase i as $\{V'_i\}_{i=1, \dots, |P|} \in \mathbb{R}^{n' \times n_{po,i}}$, and decoder parameters for subject under occurrence j as $\{W'_j\}_{j=1, \dots, |O|} \in \mathbb{R}^{n' \times n_{os,j}}$. We would like to emphasize that the decoder parameters for occurrence and subject are different and depend on the phase i and occurrence j , respectively. This is used to model the different occurrence events and transition under different phases and different subject events and transition under different occurrences. Here, we use n' to represent the dimension of the decoder vector and $n_p, n_{po,i}, n_{os,j}$ to denote the number of phases, the number of occurrences under phase i , and the number of subject codes under subject

j , respectively. p_i^t , o_j^t and s_k^t denote the phase, occurrence, and subject of flight at time t is the i -th, j -th, and k -th item in phase corpus, respectively. Since we need to compute the conditional probability across different levels, we consider to build different output vectors for same occurrence under different phases of flight. The intuition behind this is that under different phases, the same occurrence will have different impact on the aircraft and lead to different event sequences. Thus, we need to build a "conditional embedding vector" Here we use $u'_i, v'_{i,j}, w'_{j,k}$ to represent the i -th column in the decoder parameters of U' , j -th column in V'_i and k -th column in W'_j .

$$P(p_i^{t+1}|h_t) = \frac{\exp u'_i{}^T h_t}{\sum_i \exp u'_i{}^T h_t},$$

$$P(o_j^{t+1}|p_i^{t+1}, h_t) = \frac{\exp v'_{i,j}{}^T h_t}{\sum_j \exp v'_{i,j}{}^T h_t},$$

$$P(s_k^{t+1}|o_j^{t+1}, h_t) = \frac{\exp w'_{j,k}{}^T h_t}{\sum_k \exp w'_{j,k}{}^T h_t}$$

Finally, the decoder parameter θ_d will be learned from the data during the end-to-end training. Here, the overall framework of the hierarchical decoder is given in Fig. 3.

C. Prediction Using the Hierarchical Information

Here, we would like to discuss how the proposed method can be used to predict the event at the next time depending on if the higher level information is given. For example, if we do not know the phase and occurrence. The algorithm will need calculate the prediction by the summation of all possible phase p_i and occurrence o_j when doing the subject-level prediction as $\sum_{i,j} P(s_k^t|p_i, o_j, h_{t-1})P(o_j|p_i, h_{t-1})P(p_i|h_{t-1})$.

However, if the phase-level information is known as p_i^t , we can use this information in the prediction of the subject level so that the summation over the phase is not needed as $\sum_j P(s_k^t|p_i^t, o_j, h_{t-1})P(o_j|p_i^t, h_{t-1})$ so that the prediction can be more accurate. In Table II, we would like to give the prediction algorithms given different scenarios depending on whether the higher-level information such as the phase and occurrence are given.

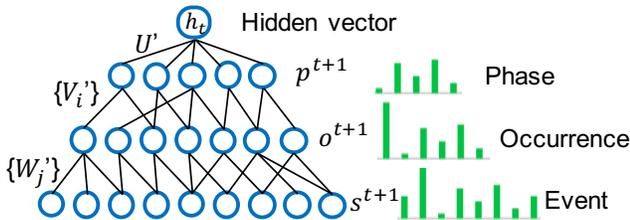


Fig. 3: Hierarchical Tree-based Decoder

IV. EXPERIMENTS

In this section, we will first introduce the NTSB data. Then, the embedding of the failure events learned from the proposed model will be discussed with the visualization. Finally, we showed a comparison of several methods to illustrate the effectiveness of the proposed event prediction model.

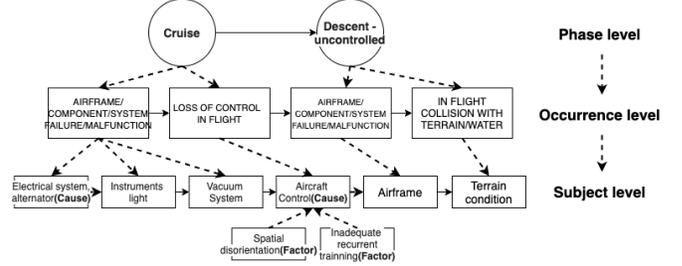


Fig. 4: Hierarchical Data Structure

A. Data Description

In this section, we will introduce the dataset and our preliminary analysis. The NTSB aviation accident database recorded information about civil aviation accidents since 1982. The data are organized in a relational database where several tables jointly describe a specific accident. Finally, we end up with 61671 accidents in 62570 aircraft. Each accident is represented with a sequence of failure events, as described in Section 3. Fig. 4 shows an example of the taxonomy of the event data, in which each event data is described using three levels from the higher level to the lower level, which are Phase, Occurrence and Subject. In the dataset, there are about 56 Phases, 58 occurrences, and 1432 subjects event. Given the large number of possible accident events especially in the subject level, the embedding algorithm can dramatically reduce the dimensionality of the event representation. The sequential failure events provide a summary of what happens during the accident. In our study, we propose a hierarchical embedding by representing the failure events from different levels using continuous vectors. We then compare the predictive performance of discrete representation with the general-purpose embedding techniques.

B. Model Deployment Procedure

Here, we would like to discuss the procedure for the proposed algorithm. The algorithm can be derived into the training and testing phase. Furthermore, we use 50045 data for training and 12511 data for testing.

During the training phase, the events $e^t = (p^t, o^t, s^t)$ for $t = 1, \dots, T$ are fully observed. In this case, we can use the likelihood function $P(\{e^t\}_{t=1}^T; \theta) = \prod_t P(e^t|h_t; \theta)$. We assume the conditional probability of the event is given as $P(e^t|h_t; \theta) = P(p^t|h_t; \theta)P(o^t|p^t, h_t; \theta)P(s^t|o^t, h_t; \theta)$. Here, the conditional probability $P(o^t|p^t, h_t; \theta)$ and $P(s^t|o^t, h_t; \theta)$ requires the hierarchical tree-structured of the events being estimated from the training data. If any

TABLE II: Prediction Strategy

	Phase Prediction	Occurrence Prediction	Subject Prediction
Unknown Phase	$P(p_i^t h_{t-1})$	$\sum_i P(o_j^t p_i, h_{t-1})P(p_i h_{t-1})$	$\sum_{i,j} P(s_k^t p_i, o_j, h_{t-1})P(o_j p_i, h_{t-1})P(p_i h_{t-1})$
Known Phase	-	$P(o_j^t p_i^t, h_{t-1})$	$\sum_j P(s_k^t p_i^t, o_j, h_{t-1})P(o_j p_i^t, h_{t-1})$
Known Occurrence	-	-	$P(s_k^t p_i^t, o_j^t, h_{t-1})$

TABLE III: Prediction Accuracy

Prediction Condition		One-hot-encoding +Markovian	Tree Embedding + Markovian	Word2Vec + LSTM	Tree Embedding + LSTM
Phase Unknown	Phase	0.4142	0.6092	0.5995	0.6287
	Occurrence	0.3767	0.5308	0.5478	0.5497
	Subject	0.2388	0.3063	0.3089	0.3377
Phase Known	Occurrence	0.7518	0.8252	0.7995	0.8297
	Subject	0.2045	0.3635	0.4196	0.4649

of the training data has transitioned between the two events, the edge of the tree will be connected.

Here, in the training phase, we apply AdamW as the optimizer and switch to SGD to ensure a stable learning curve. We select the same learning rate as 0.001.

During the testing phase, given the future $e^t = (p^t, o^t, s^t)$ is often unknown, we need to do the summation of all possible o_t to estimate the subject event s_t . One distinct advantage of the proposed hierarchical model is the ability to use high-level information, as shown in Table II. More specifically, if the future information of Phase, Occurrence, the proposed method is able to take advantage of such information to give a better prediction in the subject-level.

C. Decoder Embedding Coefficient Visualization

In this section, we would like to visualize some illustrative results of different the encoder embedding layer coefficients $U', \{V'\}, \{W'\}$ for different flight phases, occurrences, and subject codes. The hierarchical structure implies that there is only one phase embedding matrix denoted as U' , but there can be multiple occurrence embedding matrix $\{V'_i\}_{i=1, \dots, |P|}$. For example, here, we like to show the embedding matrix for the decoder occurrence under the *take-off phase* and the *landing phase* in Fig. 5. Given the embedded dimension is 30, we propose to project them into two dimensions for visualization.

Fig. 5a shows the decoding phase embedding. Similar clusters to the encoding embedding can also be found here like 560-568. We can also find some cluster are not obvious in the encoding embedding like 501-504 as the *standing phase*. 520-523 refer to the *takeoff phase*. In the decoder embedding, we can see that the *takeoff phase* is on the top right of the figure and the *landing phase* is at the bottom left of the figure, which is more meaningful to us since those two phases have very different transition diagram as illustrated in section 4.A. we further discover the conditional decoding embedding vector for occurrence. Fig. 5b denote the occurrence embedding under *takeoff phases* and fig. 5c shows the embedding under *landing phases*. We can see interesting patterns from the difference between those two visualizations. For example, we can see

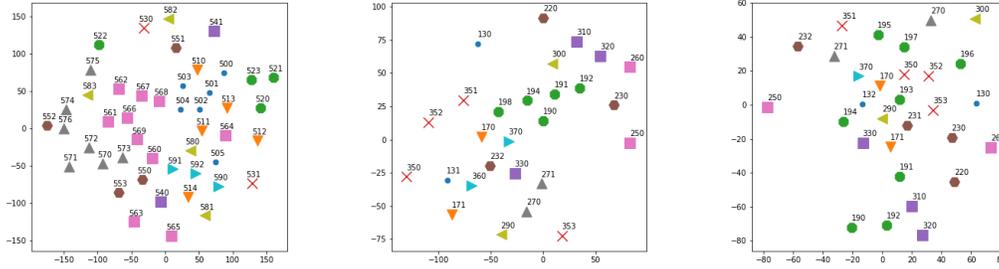
clear cluster for 190-194 (*gear collapsed*). However, there are no such clusters in the landing phase. The intuition behind this observation is that different kinds of *gear collapsed* will have a similar impact under *takeoff phase* but will have a very different impact on event sequences under *landing phase*. Similarly, 350-353 (*loss of engine power*) will have a more consistent impact on aircraft under the *landing phase* comparing to the *takeoff phase*.

D. Event Prediction Accuracy

In this section, we build a predictive model to evaluate the performance of different embedding vectors. Here, we compare four different methods. In general, we compare three different representations of the failure events, which are one-hot-encoding, word2vec encoding, and the proposed tree-based embedding. We further compare two different sequential models, namely the Markovian model and the LSTM model. Our results show that, through representing discrete events with continuous vector, we can improve the prediction accuracy with either Word2Vec or Tree Embedding. Furthermore, LSTM shows a better performance comparing to the Markovian model, given the hierarchical structure of failure events. Finally, the tree embedding shows better performance compared to the general embedding method Word2Vec in our experiments. In an aviation accident, the phase of aircraft is usually known ahead of time. Here we also present the results given the known flight phase. We show that the proposed method is able to achieve 0.8291 accuracy for occurrence level, which is a huge improvement from 0.5497.

Here we further investigate the reason we achieve better results comparing to other benchmarks without considering the multi-level information.

- The other benchmark methods, such as the Markovian models, cannot deal with the unseen event sequences. The aviation accident is usually unique and rare, so it is important to deal with events that have never occurred before. The proposed model is able to deal with the new events according to their similar sequential pattern calculated according to the event embedding.



(a) Decoder Phase Embedding, some clusters are given here: 520-523 refer to the *takeoff* phase, 570-576 refer to the *landing* phase
 (b) Decoder Occurrence Embedding under *takeoff* phase, some clusters are given here: 190-194 mean *gear collapsed*
 (c) Decoder Occurrence Embedding under *landing* phase, some clusters are given here: 350-353 mean *loss of engine power*

Fig. 5: Decoder Embedding Visualization

- Under different phases, the transition between occurrences is very different. The other benchmark methods, such as the Markovian method or the association rule-based method, can only follow a simple rule for the transition, which does not consider the high-level information. The proposed model incorporates this multi-level structure according to the hierarchical decoding layer. Through building up the conditional embedding vectors, we are able to capture different transition patterns under different phases.
- The sequential relationship is complex when we are trying to deal with multi-level event sequences. Traditional methods like association rule or Markovian models will require a lot of time to make predictions based on past event sequences. The deep learning-based model provides an efficient way to propagate the information from the beginning of the event to the end and capture its complex sequential relationship at the same time.

V. CONCLUSION

In this work, we focused on leveraging the collection of event data from the aviation accident report to improve aviation safety. More specifically, we proposed a way to predict the possible event sequences so that people are able to provide effective actions to avoid the accidents. There are three major contributions to this work. First, we proposed a hierarchical encoder-decoder framework with the domain knowledge of the taxonomy of failure event can be used in the embedding model. Second, we combined the proposed hierarchical encoder-decoder framework with the LSTM to capture the complex temporal relationship among failure events. Our results show that the hierarchical encoder-decoder combined with the LSTM has better accuracy comparing to several benchmark methods for event prediction. Furthermore, the embedding representation also provides a more meaningful result. In the future work, we would like to extend the proposed method to predict not only what type of event may happen but also when it will happen in the future.

ACKNOWLEDGMENT

The research reported in this paper was supported by funds from NASA University Leadership Initiative program (Contract No. NNX17AJ86A, Project Officer: Dr. Anupa Bajwa, Principal Investigator: Dr. Yongming Liu) and NSF DMS 1830363. The support is gratefully acknowledged.

REFERENCES

- [1] S. Wang, J. Tang, Y. Wang, and H. Liu, "Exploring implicit hierarchical structures for recommender systems." in *IJCAI*, 2015, pp. 1813–1819.
- [2] T. McCormick, C. Rudin, and D. Madigan, "A hierarchical model for association rule mining of sequential events: An approach to automated medical symptom prediction," in *Annals of Applied Statistics*, 2012.
- [3] R. K. Sahoo, A. J. Oliner, I. Rish, M. Gupta, J. E. Moreira, S. Ma, R. Vilalta, and A. Sivasubramaniam, "Critical event prediction for proactive management in large-scale computer clusters," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 426–435.
- [4] "National transportation safety board, aviation accident database," accessed 2008-05. [Online]. Available: <https://www.nts.gov/>
- [5] X. Zhang and S. Mahadevan, "Bayesian network modeling of accident investigation reports for aviation safety assessment," *Reliability Engineering & System Safety*, vol. 209, p. 107371, 2021.
- [6] C. Rudin, B. Letham, and D. Madigan, "Learning theory analysis for association rules and sequential event prediction," *Journal of Machine Learning Research*, vol. 14, no. 72, pp. 3441–3492, 2013.
- [7] J. O'Madadhain, J. Hutchins, and P. Smyth, "Prediction and ranking algorithms for event-based network data," *SIGKDD Explorations*, vol. 7, pp. 23–30, 12 2005.
- [8] Y. Li, N. Du, and S. Bengio, "Time-dependent representation for neural event sequence prediction," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Workshop Track Proceedings*, 2018.
- [9] L. Lin, L. Wen, and J. Wang, "Mm-pred: A deep predictive model for multi-attribute event sequence," in *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 2019, pp. 118–126.
- [10] L. Hu, J. Li, L. Nie, X.-L. Li, and C. Shao, "What happens next? future subevent prediction using contextual hierarchical lstm," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [11] Q. Ai, Y. Zhang, K. Bi, X. Chen, and W. B. Croft, "Learning a hierarchical embedding model for personalized product search," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2017, pp. 645–654.