Predictive Maintenance for General Aviation Using Convolutional Transformers

Hong Yang, Aidan LaBella, Travis Desell
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
20 Lomb Memorial Dr.
Rochester, New York 14623
{hy3134, apl1341, tjdvse}@rit.edu

Abstract

Predictive maintenance systems have the potential to significantly reduce costs for maintaining aircraft fleets as well as provide improved safety by detecting maintenance issues before they come severe. However, the development of such systems has been limited due to a lack of publicly labeled multivariate time series (MTS) sensor data. MTS classification has advanced greatly over the past decade, but there is a lack of sufficiently challenging benchmarks for new methods. This work introduces the NGAFID Maintenance Classification (NGAFID-MC) dataset as a novel benchmark in terms of difficulty, number of samples, and sequence length. NGAFID-MC consists of over 7,500 labeled flights, representing over 11,500 hours of per second flight data recorder readings of 23 sensor parameters. Using this benchmark, we demonstrate that Recurrent Neural Network (RNN) methods are not well suited for capturing temporally distant relationships and propose a new architecture called Convolutional Multiheaded Self Attention (Conv-MHSA) that achieves greater classification performance at greater computational efficiency. We also demonstrate that image inspired augmentations of cutout, mixup, and cutmix, can be used to reduce overfitting and improve generalization in MTS classification. Our best trained models have been incorporated back into the NGAFID to allow users to potentially detect flights that require maintenance as well as provide feedback to further expand and refine the NGAFID-MC dataset.

Introduction

In the domain of aviation, especially for small scale general aviation fleets, aircraft maintenance is performed with fixed schedules or after some maintenance issue is detected during operation of an aircraft. *Predictive maintenance* techniques can be performed to reduce cost, improve machinery performance and life, as well as mitigate risk and increase safety. The majority of published literature covers non neural network methods (Carvalho et al. 2019). Machine learning presents the potential to predict maintenance issues by measuring anomalies or degradation of multivariate time series (MTS) sensor data; however this has been limited by the proprietary nature of most flight data, with the further issue of acquiring the data necessary to label flight data with and without specific maintenance issues.

While the abundance of multivariate temporal data has enabled significant advances in MTS analysis for a wide variety of fields, current literature lacks an evaluation of MTS methods for non synthetic, extremely long sequences (greater than 1024 time steps) from large labeled datasets (more than 5000 datapoints) (Fawaz et al. 2019). This paper utilizes data from the National General Aviation Flight Information Database (NGAFID) and the MaintNet project to create a new large scale labeled MTS benchmark, the NGAFID Maintenance Classification dataset (NGAFID-MC), with over 7,500 labeled flight sensor data files[1], for development of predictive maintenance systems for aviation.

Using this dataset, our results show that previous MTS classification methods face great difficulty in classifying pre and post maintenance flights. We also demonstrate that a new Convolutional Multiheaded Self Attention architecture can better capture complex temporally distant relationships within NGAFID-MC and leverage them for better classification performance and computational efficiency. We also demonstrate the need for robust augmentations and introduce a set of MTS augmentations that improve generalization. We provide a Google Colab Notebook for anyone to fully replicate the results of our experiments[2]. Finally, our best trained models have been reincorporated into the NGAFID to inform users and collect their feedback to further refine the models and expand the NGAFID-MC dataset.

Related Work

Several methods have been developed for MTS classification, for a review see (Fawaz et al. 2019). Notable non-deep learning methods include distance based k-nearest neighbors by (Orsenigo and Vercellis 2010) and Dynamic Time Warping KNN by (Seto, Zhang, and Zhou 2015). For deep learning methods, well performing MTS classifiers tend to utilize some combination of Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) methods, *e.g.* (Karim et al. 2017), or Temporal CNN (TCNN) methods (Wang, Yan, and Oates 2017). However, RNN methods struggle with long sequences due to the vanishing gradient problem (Le and Zuidema 2016). TCNN methods perform well for MTS classification (Assaf et al. 2019), but may struggle

[1] https://www.kaggle.com/hooong/ngafid-mc-20210917
[2] https://tinyurl.com/b35mxv98

when relevant features are temporally sparse and related.

Transformer models have been used in sequence tasks, such as NLP by (Devlin et al. 2018) and MTS prediction by (Zhou et al. 2021). They do not suffer the vanishing gradient problem described by (Le and Zuidema 2016), allowing them to learn more temporally distant relationships. Application of transformer models and their underlying Multiheaded Self Attention (MHSA) mechanisms may lead to performance gains compared to RNNs.

(Fawaz et al. 2019) notes that time series augmentation lacks a thorough study, compared to NLP and Computer Vision. Augmentations techniques in Computer Vision, such as cutmix (Yun et al. 2019), may be applicable to MTS data.

The datasets used by (Fawaz et al. 2019) do not exceed 1024 timesteps, except the WalkVsRun dataset consisting of only 28 training and 16 test examples. To the authors' knowledge, there are no MTS datasets that are not simulated, have greater length than 1024, and have labeled examples greater than 5000. Datasets meeting this criteria provide more realistic benchmarks for many real world applications, especially those related to engineering systems and predictive maintenance.

Dataset and Data Collection

The NGAFID serves as a repository for general aviation flight data, with a web portal for viewing and tracking flight safety events for individual pilots as well as for fleets of aircraft (Karboviak et al. 2018). The NGAFID currently contains over 900,000 hours of flight data generated by over 780,000 flights by 12 different types of aircraft, provided by 65 fleets and individual users, resulting in over 3.15 billion per second flight data records across 103 potential flight data recorder parameters. Five years of textual maintenance records from a fleet which provided data to the NGAFID have been clustered by maintenance issue type and then validated by domain experts for the MaintNet project (Akhbardeh, Desell, and Zampieri 2020). Flights were extracted from the NGAFID and labeled as before or after the date of the maintenance action, creating a MTS dataset for training predictive maintenance models.

MaintNet's maintenance record logbook data was clustered into 39 different maintenance issue types. Because some issues occurred very rarely, this work focused on the two largest clusters, representing the most common maintenance issues: cluster 28 (C28): intake gasket leak/damage and cluster 37 (C37): rocker cover loose/leak/damage. The C28 and C37 clusters contain 1674 and 1248 maintenance records, respectively. Using the tail number from these maintenance records, the five flights preceding any of these maintenance records were exported from the NGAFID to represent flight data relating to those maintenance issues. To provide a robust set of "good" flights without maintenance issues to compare these against, the five flights after the maintenance issues were exported as well, unless they were within 5 flights of any other maintenance issue. Flights shorter than 30 minutes were excluded as these are typically do not involve any actual flight. Flights were then further filtered within 2 days of maintenance (before and after). As the maintenance records only provided a day (and not a time) of action, flights occurring on the same day as maintenance were excluded as it was not possible to determine if they occurred before or after maintenance.

This resulted in a benchmark dataset containing 7,505 flight data files representing 11,500 hours of Cessna 172S flight data, with each flight data file in this dataset consisting of data from 23 sensors (internal, external and operational sensors, *e.g.*, engine RPM, oil temperature, oil pressure, gasket temperature, airspeed, pitch, roll, outside air temperature) recorded every second, with each flight labeled as pre or post maintenance. Flights are split for the two maintenance issues resulting in 1432 pre and 984 post examples for C37 and 2814 pre and 2275 post examples for C28.

Background

A major goal of this work is to be able to classify flights as problematic (leading to some maintenance issue), or in good condition (post maintenance). Three factors make this dataset challenging. First is the sequence length, often exceeding 3600 time steps. Second is the nature of the prediction task, where the goal is to detect features relevant for classification. Third is the significant impact of unobservable variables, such as pilot actions, on the engine outputs.

To formalize the problem, we seek to predict the probability that a time series was generated by a pre or post maintenance flight given the flight sensor data. This can be expressed as $P(Y_i|X_i)$. We have access to the variable X_{imt} as a matrix containing the flight sensor data, with imt representing the ith flight's mth variable at timestep t. Y_i represents the ith flight's pre or post maintenance state as 1 and 0, respectively. U_{it} represents the pilot's actions for the ith flight's timestep t. This unknown variable U is significant because it changes our understanding of the function generating X to $f(U_i, Y_i) = X_i$. A pilot's actions can impact X_{imt} more than maintenance state of the aircraft.

We cannot construct a model to to predict $X_{im(t+1)}$ using only the past timesteps of X_i due to the impact of U_i. Similarly, a compressed representation $c(X)$ may be useless for classification because it must first explain variance caused by U. The authors believe that non-deep learning methods will struggle to perform well in these conditions.

This dataset provides an exciting challenge compared to industrial datasets, such as power plant data, because it measures a dynamic system that changes arbitrarily in a largely uncontrolled and inconsistent environment. Routine flight operations, such as landing and takeoff, can vary significantly from flight to flight due to the experience of the pilot, the weather, and wind conditions. We hope this dataset can serve as a challenging benchmark for MTS classification.

Model Architecture and Training

Augmentation To address the limitations of the size of this dataset, we looked into augmentations for MTS data. We considered only basic domain augmentation methods based on the taxonomy proposed by (Wen et al. 2020), as advanced domain augmentations are too complex, requiring one to train generative models. Basic time domain methods, as described by (Le Guennec, Malinowski, and Tavenard 2016)

and (Wen et al. 2020), include window slicing (training on slices of the MTS) and window warping (reducing or extending the length of a segment of the MTS). These methods were not seen to be applicable as window slicing should fail if features for classification are temporally distant and infrequent, and window warping may not be applicable if the data is not sinusoidal in nature, which this data is not.

The authors of this paper decided to explore new basic domain augmentations, inspired by highly effective augmentations for image classification. We consider the ideas cutout from (DeVries and Taylor 2017), mixup from (Zhang et al. 2017), and cutmix from (Yun et al. 2019). To our knowledge, this paper is the first to evaluate the these augmentations on MTS, due to their absence in the surveys on MTS augmentation by (Wen et al. 2020) and (Iwana and Uchida 2021).

Temporal cutout selects a random time segment and set of channels from a MTS and sets selected values to 0. Temporal cutmix selects a random time segment from the first MTS and a random time segment of a random second MTS of any label. It then selects a random set of channels and replaces the first MTS's segment's channel values with those of the second. Temporal mixup multiplies a randomly selected set of channels in the first MTS by a value, m, and then adds all values from the same set of channels from a randomly selected second MTS of any label multiplied by $1 - m$.

Convolutional Multi-Headed Self Attention Multi-Headed Self Attention (MHSA) modules were popularized by (Devlin et al. 2018) for usage in Natural Language Processing and by (Dosovitskiy et al. 2020) for Computer Vision, but published usage of MHSA in MTS data is less common than the previous two applications. Both (Song et al. 2018) and (Rußwurm and Körner 2020) use MHSA for MTS classification, but neither implements a 1D convolution followed by MHSA. (Karim et al. 2019) utilizes attention mechanisms in an Attention-LSTM network, but not MHSA. The benefits of a low level convolution prior to MHSA is demonstrated by (Gulati et al. 2020), where convolutions can capture basic local relationships with high efficiency while MHSA handles global relationships.

At the time of writing, the authors are not aware of any publication that evaluates a convolutional MHSA model for MTS classification. This model implements attention layers that mimic the functionality of the encoder layers present in BERT (Devlin et al. 2018). Instead of token embeddings, the model generates sequence embeddings with the use of 1D convolutions along the temporal dimension. These learnable sequence embeddings capture local relationships and to compress the MTS to a shorter length.

MHSA modules offer significant benefits over LSTMs when applied to MTS. In particular, (Zhou et al. 2021) has shown the capacity for MHSA to model long term relationships in time series data. Furthermore, when applied to longer sequences, MHSA avoids the problems associated with a vanishing gradient as described by (Le and Zuidema 2016). Not only does it better model long term relationships, there is a significant computational efficiency over RNNs.

The Conv-MHSA model evaluated in this paper (see Figure 1) uses a series of 1D convolutions to reduce the temporal resolution from 4096 to 512 and then employs 4 stacked MHSA encoder layers with 8 heads each and 64 dense units per head. The output is globally average pooled and fed to a dense layer for classification.

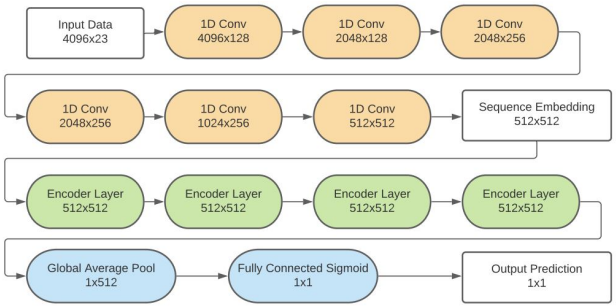

Figure 1: Layers and output shapes of the Conv-MHSA model. The first dimension represents time and the second represents channels. Note that white the boxes do not perform an operation, but mark significant states in the network.

Convolutional Long Short Term Memory Networks (Keren and Schuller 2016) present a 1 dimensional convolutional LSTM as an enhancement to the traditional RNN. By using a 1 dimensional convolution, it is possible to extract features from the sequence before the LSTM layers and reduce MTS temporal resolution.

We consider two Conv-LSTM models. The first, referred to as Conv-LSTM, utilizes the same series of 1D convolutions as the Conv-MHSA model, but instead employs 4 stacked Bidirectional LSTMs with 512 units. The output is globally average pooled and fed to a dense layer for classification. The second Conv-LSTM is referred to as EX-Conv-LSTM, which utilizes 2 additional 1D convolutions before the stacked Bidirectional LSTMs to further reduce the temporal resolution to 128.

Convolutional GRU Variational Auto Encoders Variational Autoencoders (VAE) were popularized by (An and Cho 2015) for anomaly detection. VAE's assume that a system's observable outputs X can be described via a vector or embedding E, generated by the encoder component of the VAE model. When X cannot be described via a model generated embedding E, it indicates anomalous activity. The ability to describe X via E is based on the ability of a decoder model to reconstruct X using only E and is measured as the reconstruction error. VAE learn their embeddings as a Gaussian distribution, with Kullback–Leibler divergence (KLD) for regularization. (An and Cho 2015) shows better performance for VAEs over standard Autoencoders.

GRU based VAE models have been employed by (Guo et al. 2018) for anomaly detection in MTS data. For classification, this approach trains on within class data (post maintenance) with the expectation that out of class data (pre maintenance) will have greater reconstruction error.

We implemented a VAE-Conv-GRU that uses 1D convolutions to reduce the temporal resolution to 256, followed by a bidirectional GRU (BD-GRU) with 256 units, then a

Model	Step Time in ms	Parameters
C.MHSA	50	7.9M
C.LSTM	800	24.7M
EX C.LSTM	220	28.4M
VAE-Conv-GRU	130	18.3M

Table 1: Approximate Training Step Time in Miliseconds

1D convolution to reduce the temporal resolution to 128, followed by another BD-GRU with 512 units. An embedding of 512 mixture Gaussian distributions and 8 mixtures per distribution is generated from the BD-GRU outputs, regularized via KLD. The decoder structure matches the encoder, with 1D Transposed Convolutions for the purpose of expanding the temporal resolution. Augmentations were not used.

Training Setup All results reported were generated using a Google Colab instance with a v2-8 TPU. All models were trained for 30 epochs using a batch size of 32, with 5-fold cross validation. Steps per epoch are 250 for MHSA and LSTM models, 1000 for VAE models, and 500 for extended training LSTM models. Flights are truncated to the last 4096 time steps and padded to be of the same size. To ensure the validation data is a good measure of generalized performance, the validation data is only composed of flights from tail numbers (unique identifier for planes) not present in the training data. Classification models used an Adam optimizer with a decaying learning rate starting at 1e-5 for MHSA and 2e-5 for LSTM and VAE models used an Adam optimizer with a decaying learning rate starting at 1e-4. Each augmentation (temporal cutout, cutmix, and mixup) was performed on a MTS with a 40% chance. The time segment length for cutout and cutmix was selected uniformly at random between 64 and 512. Each channel had a 30% chance of being selected for cutout and cutmix. For temporal mixup, m was selected uniformly at random between 0.6 and 0.9, and it was applied to all time steps, with channels being selected with a 40% chance.

Results

Computational Efficiency We observe significant computational advantages in the training of the Conv-MHSA compared to all other models. When using a TPU, the training step time (time to train on 1 batch) of Conv-MSHA is at least 4x faster than Extra-Conv-LSTM and at least 15x faster than Conv-LSTM. The results are summarized in Table 1.

Some of these advantages in step time could be caused by TPUs, which utilize matrix multiplication units (MXU's). Performance may differ on GPU systems.

Classification Performance We evaluate each model's Area Under the Curve score for Precision-Recall (PR) and Receiver Operating Characteristic (ROC). These threshold independent metrics better measure generalized model performance than accuracy. Accuracy (ACC) is excluded from analysis because it depends on defining a threshold for predictions, which may be misleading due to class imbalance. Binary Cross Entropy loss in also considered as a metric to evaluate model overconfidence in wrong predictions. Results for VAE-Conv-GRU models are excluded from the table due to poor performance. See Table 2.

	Model Type	A	Loss	ROC	PR	ACC
C28	C.LSTM	Y	0.630	0.701	0.654	0.653
		N	0.617	0.742	0.697	0.685
	C.LSTM+	Y	0.623	0.730	0.644	0.691
		N	0.613	0.757	0.711	0.694
	C.MHSA	**Y**	**0.528**	**0.826**	**0.802**	0.744
		N	0.557	0.819	0.792	**0.751**
	EX C.LSTM	Y	0.612	0.725	0.678	0.667
		N	0.614	0.755	0.713	0.694
	EX C.LSTM+	Y	0.608	0.764	0.699	0.718
		N	0.612	0.785	0.737	0.733
C37	C.LSTM	Y	0.643	0.674	0.567	0.655
		N	0.679	0.553	0.489	0.596
	C.LSTM+	Y	0.635	0.723	0.639	0.693
		N	0.644	0.711	0.618	0.683
	C.MHSA	**Y**	**0.601**	**0.775**	**0.711**	**0.723**
		N	0.680	0.559	0.485	0.590
	EX C.LSTM	Y	0.632	0.708	0.620	0.677
		N	0.640	0.709	0.608	0.680
	EX C.LSTM+	Y	0.639	0.731	0.643	0.699
		N	0.651	0.714	0.619	0.681

Table 2: Mean of the best metrics for each configuration. LSTM + models are trained for 500 steps per epoch. C. stands for Conv. A stands for augmented.

Results indicate that Conv-MHSA models consistently perform better than Conv-LSTM models by a wide margin. Even when Conv-LSTM models are given twice the number of training steps, they fail to reach the performance of MHSA models.

Classification using VAE-Conv-GRU While the VAE-Conv-GRU model is capable of achieving a validation Root Mean Squared error of 0.0338, it cannot predict pre or post maintenance. With mean squared error as the reconstruction loss for comparing within class and out of class examples, the PR-AUC and ROC-AUC values never exceed 0.55.

Discussion

Temporally Distant Attention To explore the question as to why MHSA can achieve better performance on this dataset compared to RNNs, it is important to observe how the various heads attend to different positions of the sequence. Figure 2 is a visualisation of the 4 MHSA layers with multiple input datapoints. We can clearly observe in sample 0 that some layers are attending to time steps that are 300 units apart. An RNN model may have great difficulty in propagating information from time step 50 to time step 400 due to memory degradation and vanishing gradients. MHSA allows any time step to attend to any other time step and better capture temporally distant relationships.

To further show that the relationship between temporally distant features is necessary for classification, we attempted to train a Short-LSTM network using randomly sampled slices 128 time steps long. This network uses 4 stacked 512 unit bidirectional LSTMs followed by global average pool-

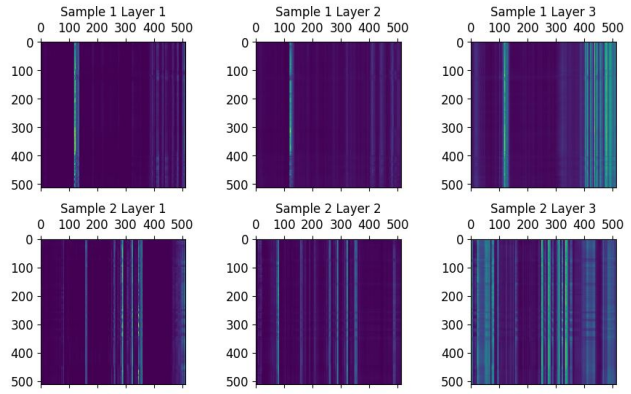

Figure 2: Attention maps illustrate how each time step attends to other timesteps in MHSA. This shows 3 MHSA Layers for 2 different datapoints from validation data. Y Axis represents Query and X axis represent Key. Sample 2 is positive and sample 1 is negative. Bright sections are important timesteps that the model focuses on.

ing and a dense sigmoid layer for classification. This Short-LSTM network did not perform significantly better than a fully random predictor, outputting random floating point values between 0 and 1. This demonstrates that random sub samples from the overall MTS is not sufficient.

Augmentation The 3 augmentations of cutout, mixup, and cutmix, have similar functionality as dropout, described by (Srivastava et al. 2014). While it may seem counter intuitive to generate unrealistic sequences, these augmentations penalize the model for memorizing a small subset of time steps by removing or modifying them. Like their computer vision equivalents, these augmentations help models learn more resilient representations and improve generalization.

Augmentations are particularly important for Conv-MHSA networks, which are prone to overfitting on small datasets. Conv-LSTM networks do not overfit and may not benefit from augmentation. Results from experiments on Conv-MHSA models show a small advantage in the mean of all metrics when training on the C28 dataset, but a significant advantage when training on the C37 dataset. These differences are significant, such that Conv-MHSA models trained on C37 without augmentation perform not significantly better than random guessing. This is most likely caused by a difference in the dataset size, where C37 is about half the size of C28. This difference may also be caused by a difference in the nature of the data, where it is possible that C28 is easier to generalize on than C37. Figure 3 shows the Conv-MHSA overfitting when training without augmentations on the C37 dataset.

VAE and Reconstruction Loss Figure 4 shows that the reconstruction loss is the same for both classes. This likely because a significant portion of the variance in X is caused by an unobservable variable U. Any VAE model would first seek to learn how U impacts X. Based on the analysis of MHSA on this dataset, there may be only a few, short segments of the MTS that are actually useful for classification. This suggests that VAE methods may struggle.

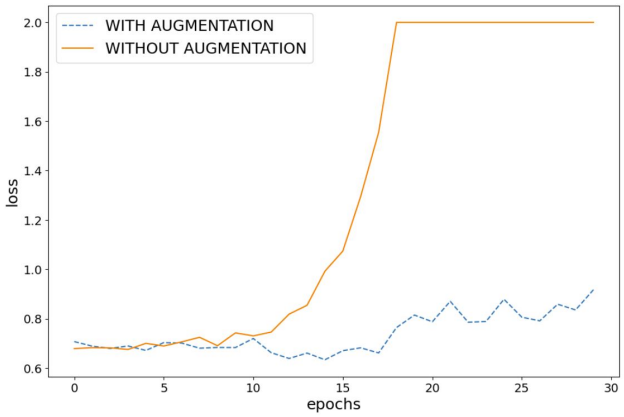

Figure 3: Validation Loss by epoch for Conv-MHSA model on the C37 dataset.

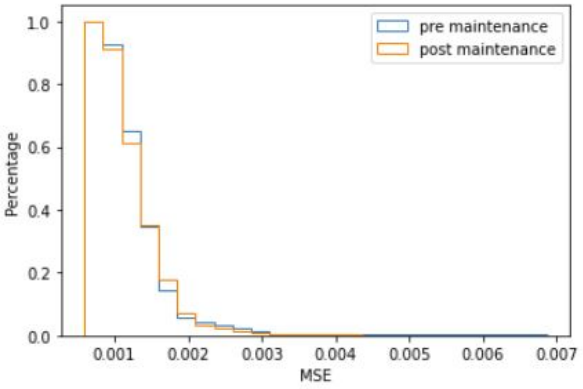

Figure 4: The Y axis indicates percentage of validation datapoints having more MSE than the number in X axis. The orange and blue lines represent pre and post maintenance, respectively. The distributions show no significant difference.

Future Research The NGAFID-MC dataset can be used to evaluate a wide variety of models and approaches, such as TCNN (Assaf et al. 2019) and dynamic time warping (Seto, Zhang, and Zhou 2015). Further studies can be performed on the full flight sequences, rather than only the last 4096 seconds of the flight. Future work should also evaluate mutliclass classification to identify which issue is present. We also intend to expand the NGAFID-MC dataset with more maintenance issue cluster types, as well as refinements based on user annotations and as additional maintenance records are received.

The Conv-MHSA architecture performs much better than Conv-LSTM models on this dataset and it would be interesting to evaluate this on other datasets. It is also plausible that we can improve Conv-MHSA architecture by incorporating memory efficient methods described by (Kitaev, Kaiser, and Levskaya 2020). Additional work should be done on alternative loss functions for MTS classification, such as focal loss

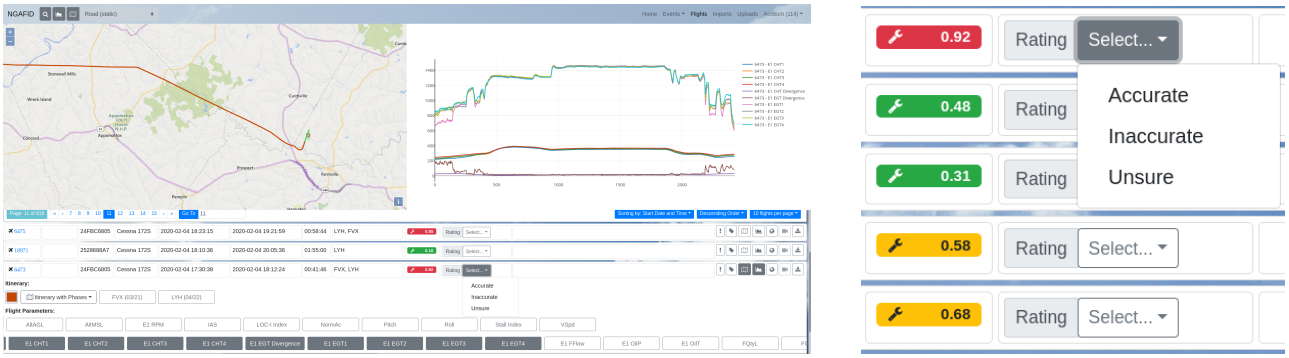

Figure 5: Screenshots showing the integration of the maintenance prediction models into the NGAFID user interface.

(Lin et al. 2017) and label smoothing (Szegedy et al. 2016).

Cutout, mixup, and cutmix augmentations should be evaluated against other MTS augmentation methods and models. Due to the limited size of many MTS datasets and the cost to acquire data, further study into augmentation can increase the viability of MTS classification methods for general use.

Limitations There may be mislabeled datapoints due to the nature of airplane maintenance. A reported maintenance issue may not be fully fixed or an issue was falsely identified by the pilot. If given resources, the authors would like to construct a small and rigorously annotated test set of data (1000 examples) with the help of domain experts. Additionally, the flights which occured on the day of maintenance were not included, these will be included in the future as they are annotated by domain experts.

NGAFID Deployment and Integration

The NGAFID provides a set of utilities for Flight Data Monitoring (FDM), which allow users to access the per-second time series data and perform various analytics. We added additional functionality to calculate and display the probability that a flight may require maintenance for the Cessna 172S aircraft type for flights exceeding 30 minutes (see Figure 5). This includes a feedback system was created to give users the ability to rate the accuracy of $P(Y_i|X_i)$ using a three-point scale (accurate, inaccurate or unsure), based on their knowledge of aviation and aircraft maintenance. This allows users to provide valuable feedback and labeled data for refining and improving future models.

However, there are infrastructure challenges that need to be addressed before NGAFID can provide real time predictive maintenance alerts to improve safety and reduce costs. The main obstacle is the lack of wireless flight data transmission (WFDT), which is more common in commercial aviation settings. The current data import process for the NGAFID occurs weekly and requires ground crews to manually extract and upload the data. NGAFID partner fleets are in the process of deploying WFDT systems that will allow the NGAFID to perform real time predictive maintenance, as the WFDT systems can upload data immediately after an aircraft lands and returns to the hangar.

Conclusion

We demonstrate the challenging nature of the NGAFID-MC dataset and its value for assessing various MTS approaches. While some datasets exceed NGAFID-MC in terms of datapoints or sequence length, the authors are not aware of any dataset that has both greater datapoints and sequence length. The authors are also not aware of any other MTS dataset that tracks a dynamic system that changes arbitrarily in a largely uncontrolled and inconsistent environment. Furthermore, we demonstrate that this dataset contains temporally distant relationships that previous MTS classification methods struggle with. We hope that the difficulty of this dataset will inspire new and better methods for MTS classification.

We also introduce a more computationally efficient and performant architecture, the Conv-MHSA. This architecture can better capture temporally distant relationships in long sequences and it does so with at much greater computational efficiency than RNN methods. We also show that cutmix, cutout, and mixup augmentations can significantly improve generalization.

The ability to differentiate between pre and post maintenance flights leads can provide a significant benefit to the domain of general aviation. Early detection of maintenance issues has the potential to reduce long term maintenance costs by catching issues before they cause more serious problems. By detecting the need for maintenance one or two days prior to maintenance, we can minimize the amount of flight hours that a pilot spends on compromised aircraft, leading to increased safety. We have already incorporated preliminary models for maintenance classification for NGAFID, which will allow us to gather feedback from users to further refine and improve the early maintenance issue detection system. We hope that these tools will lead to increased safety and reduced costs for general aviation.

References

Akhbardeh, F.; Desell, T.; and Zampieri, M. 2020. MaintNet: A Collaborative Open-Source Library for Predictive Maintenance Language Resources. In *Proceedings of the 28th International Conference on Computational Linguistics: System Demonstrations*, 7–11. Barcelona, Spain (Online): International Committee on Computational Linguistics (ICCL).

An, J.; and Cho, S. 2015. Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE*, 2(1): 1–18.

Assaf, R.; Giurgiu, I.; Bagehorn, F.; and Schumann, A. 2019. Mtex-cnn: Multivariate time series explanations for predictions with convolutional neural networks. In *2019 IEEE International Conference on Data Mining (ICDM)*, 952–957. IEEE.

Carvalho, T. P.; Soares, F. A. A. M. N.; Vita, R.; da P. Francisco, R.; Basto, J. P.; and Alcalá, S. G. S. 2019. A systematic literature review of machine learning methods applied to predictive maintenance. *Computers and Industrial Engineering*, 137: 106024.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Fawaz, H. I.; Forestier, G.; Weber, J.; Idoumghar, L.; and Muller, P.-A. 2019. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4): 917–963.

Gulati, A.; Qin, J.; Chiu, C.-C.; Parmar, N.; Zhang, Y.; Yu, J.; Han, W.; Wang, S.; Zhang, Z.; Wu, Y.; et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Guo, Y.; Liao, W.; Wang, Q.; Yu, L.; Ji, T.; and Li, P. 2018. Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach. In *Asian Conference on Machine Learning*, 97–112. PMLR.

Iwana, B. K.; and Uchida, S. 2021. An empirical survey of data augmentation for time series classification with neural networks. *Plos one*, 16(7): e0254841.

Karboviak, K.; Clachar, S.; Desell, T.; Dusenbury, M.; Hedrick, W.; Higgins, J.; Walberg, J.; and Wild, B. 2018. Classifying aircraft approach type in the national general aviation flight information database. In *International Conference on Computational Science*, 456–469. Springer.

Karim, F.; Majumdar, S.; Darabi, H.; and Chen, S. 2017. LSTM fully convolutional networks for time series classification. *IEEE access*, 6: 1662–1669.

Karim, F.; Majumdar, S.; Darabi, H.; and Harford, S. 2019. Multivariate LSTM-FCNs for time series classification. *Neural Networks*, 116: 237–245.

Keren, G.; and Schuller, B. 2016. Convolutional RNN: an enhanced model for extracting features from sequential data. In *2016 International Joint Conference on Neural Networks (IJCNN)*, 3412–3419. IEEE.

Kitaev, N.; Kaiser, Ł.; and Levskaya, A. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Le, P.; and Zuidema, W. 2016. Quantifying the vanishing gradient and long distance dependency problem in recursive neural networks and recursive LSTMs. *arXiv preprint arXiv:1603.00423*.

Le Guennec, A.; Malinowski, S.; and Tavenard, R. 2016. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Orsenigo, C.; and Vercellis, C. 2010. Combining discrete SVM and fixed cardinality warping distances for multivariate time series classification. *Pattern Recognition*, 43(11): 3787–3794.

Rußwurm, M.; and Körner, M. 2020. Self-attention for raw optical satellite time series classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169: 421–435.

Seto, S.; Zhang, W.; and Zhou, Y. 2015. Multivariate time series classification using dynamic time warping template selection for human activity recognition. In *2015 IEEE Symposium Series on Computational Intelligence*, 1399–1406. IEEE.

Song, H.; Rajan, D.; Thiagarajan, J. J.; and Spanias, A. 2018. Attend and diagnose: Clinical time series analysis using attention models. In *Thirty-second AAAI conference on artificial intelligence*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Wang, Z.; Yan, W.; and Oates, T. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, 1578–1585. IEEE.

Wen, Q.; Sun, L.; Yang, F.; Song, X.; Gao, J.; Wang, X.; and Xu, H. 2020. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*.

Yun, S.; Han, D.; Oh, S. J.; Chun, S.; Choe, J.; and Yoo, Y. 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6023–6032.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; and Zhang, W. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI*.