ISTANBUL TECHNICAL UNIVERSITY ★ FACULTY OF AERONAUTICS AND ASTRONAUTICS

ARTIFICIAL NEURAL NETWORKS IN AERODYNAMICS

GRADUATION PROJECT

Oğuz ÖZDOĞAN

Department of Aeronautical Engineering

Thesis Advisor: Prof. Dr. Fırat Oğuz EDİS

JULY, 2020

ISTANBUL TECHNICAL UNIVERSITY ★ FACULTY OF AERONAUTICS AND ASTRONAUTICS

ARTIFICIAL NEURAL NETWORKS IN AERODYNAMICS

GRADUATION PROJECT

Oğuz ÖZDOĞAN
(110150024)

Department of Aeronautical Engineering

Thesis Advisor: Prof. Dr. Fırat Oğuz EDİS

JULY, 2020

Oğuz Özdoğan, student of ITU Faculty of Aeronautics and Astronautics student ID **110150024**, successfully defended the **graduation** entitled "**ARTIFICIAL NEURAL NETWORKS IN AERODYNAMICS**", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Prof. Dr. Fırat Oğuz EDİS** Istanbul Technical University	.....
Jury Members :	**Prof. Dr. N. L. Okşan Çetiner Yıldırım** Istanbul Technical University	.....
	Doç. Dr. Ayşe Gül Güngör Istanbul Technical University	.....

Date of Submission : 13 July 2020
Date of Defense : 22 July 2020

To my family,

FOREWORD

I would like to thank my advisor Prof. Dr. Fırat Oğuz EDİS for introducing me the problem and guiding me whenever i need.

I am also greteful to Prof. Dr. N. L. Okşan Çetiner YILDIRIM for providing the datasets used in this study.

Also, I thank to Buğra Soysal, Okan Maden, Sadık Utku Gülek and Sercan Öztürk, my colleagues I have worked on the project together, for their support and help.

July 2020 Oğuz ÖZDOĞAN

TABLE OF CONTENTS

Page

ABBREVIATIONS

CFD	**:** Computational Fluid Dynamics
ANN	**:** Artificial Neural Network
RNN	**:** Recurrent Neural Network
LSTM	**:** Long Short-Term Memory
AoA	**:** Angle of Attack
APE	**:** Absolute Percetage Error
MSE	**:** Mean Squared Error
API	**:** Application Programming Interface
Adam	**:** Adaptive Moment Estimation
tanh	**:** Hyperbolic Tangent

LIST OF TABLES

Page

LIST OF FIGURES

Page

ARTIFICIAL NEURAL NETWORKS IN AERODYNAMICS

SUMMARY

Nowadays, the progress in artificial intelligence is spreading rapidly to all fields of natural sciences and engineering, therefore it is not difficult to research and find new topics to create creative ideas with these tools. In addition to traditional methods in aerodynamics, new oppurtunities have been arised for studies that could not be done before, thanks to the advanced algorithms and various architectures of artificial neural networks (ANN).

In this graduation project study, using the data of an experiment in which the transient gust effect on a wing model is examined with various configurations; with the special ANN type Long Short-Term Network (LSTM) model, predictions were made and compared with the actual data. Gust is a very interesting topic in aerodynamics because it is related to turbulence. Therefore, it will be good both theoretically and practically to estimate gust behavior. The reason for choosing LSTM as the ANN type in this study is that the basis of the study is the time series forecasting. LSTMs are customized version of Recurrent Neural Networks (RNN) for time series forecasting. The vanishing gradient problem of RNNs disappeared in LSTMs.

Many parameters must be selected for the creation of the LSTM model. Optimizing each of these parameters is very difficult because the parameters perform based on specific datasets. However, some important parameters have been optimized. At the end, the model has been tested in different configurations for different parameters and has been compared with real data. Variation of angle of attack, change of y-shift parameter, forces in the x and y direction and moving average method were used as configurations. It was observed that the angle of attack and y-shift did not have a decisive influence on the predictive power of the model. Note that the force prediction in the direction of y is less successful than the force prediction in the direction of x. Also, moving average applied model gave a positive result.

The results are generally satisfactory and it seems that new applications will emerge as the LSTM applications are recognized by the researchers in aerodynamics. This study showed that ANNs and LSTM networks should be used more frequently as a side tool in aerodynamics.

AERODİNAMİKTE YAPAY SİNİR AĞLARI

ÖZET

Yapay zekadaki hızlı ilerlemenin doğa bilimleri ve mühendislikteki bütün alanlara hızla yayıldığı bugünlerde, bu araçlar yardımıyla araştırma yapmak ve yaratıcı fikirler ortaya atacak yeni konular bulmak pek zor değildir. Aerodinamikteki geleneksel metotların yanında, yapay sinir ağlarının (ANN) gelişmiş algoritmaları ve türlü mimarileri sayesinde daha önce yapılamayan çalışmalar için alan açılmıştır.

Bu bitirme projesi çalışmasında, bir kanat modeli üstündeki geçici gust etkisinin çeşitli konfigürasyonlarla incelendiği bir deneyin verileri kullanılarak; özel bir ANN tipi olan Long Short-Term Network (LSTM) modeli ile tahminler yapılmış ve gerçek verilerle karşılaştırması yapılmıştır. Gust konusu aerodinamikte çok ilginç bir konudur çünkü türbülansla ilişkilidir. Bu yüzden gust davranışının tahminini yapabilmek, hem teorik hem de pratik açıdan iyi olacaktır. Bu çalışmada ANN tipi olarak LSTM seçilmesinin sebebi ise, çalışmanın esasının zaman serisi tahmini olmasıdır. LSTM'ler Recurrent Neural Networks (RNN)'lerin zaman serisi tahmini için özelleşmiş bir halidir. RNN'lerin kaybolan gradyan (vanishing gradient) problemi LSTM'lerde ortadan kalkmıştır.

Daha önceki aerodinamik ve akışkanlar mekaniği uygulamalarında da başarılı olan LSTM, bu proje çalışmasında da tatmin edici sonuçlar vermiştir. LSTM modelinin kurulması için çokça parametrenin seçilmesi gerekir. Bu parametrelerin her birinin optimize edilmesi çok zordur çünkü parametreler spesifik veri setlerine bağlı olarak performans gösterirler. Ancak bazı önemli parametreler optimize edilmiştir.

Bu çalışmada, model farklı parametreler için farklı konfigürasyonlarda test edilmiştir ve gerçek verilerle karşılaştırılıp hata analizi sunulmuştur. Konfigürasyon olarak hücum açısının değişimi, y kayması parametresinin değişimi, x ve y yönündeki kuvvetler ve hareketli ortalama metodu kullanılmıştır. Hücum açısının modelin tahmin gücü üzerinde belirleyici bir etkisi olmadığı görülmüştür. y yönündeki kuvvetin tahmininin ise x yönündeki kuvvetin tahminine kıyasla daha az başarılı olduğu kaydedilmiştir. Y kayması konfigürasyonu da tıpkı hücum açısı gibi belirleyici olmamıştır. Hareketli ortalama uygulanan model ise olumlu bir sonuç vermiştir. Model y kayması ve hücum açısından bağımsız olarak başarı göstermiştir.

Sonuçlar genel anlamda tatmin edicidir ve görünen odur ki aerodinamikte LSTM uygulamaları araştırmacılar tarafından fark edildikçe yeni uygulama alanları ortaya çıkacaktır. Bu çalışma, yapay sinir ağlarının ve LSTM mimarisinin aerodinamikte yan araç olarak daha sık kullanılması gerektiğini göstermiştir.

1. INTRODUCTION

Computational Fluid Dynamics (CFD) and experimental methods have traditionally been used in fluid mechanics and aerodynamics. Although these methods will still remain useful tools for researchers, there is a need for new tools that will reduce time and cost due to certain disadvantages of these methods. With the development of computing power, Artificial Neural Networks (ANN) algorithms and new statistical analysis tools day by day, CFD and experimental methods have new tools to accompany them. Because of the new tools, research fields in fluid mechanics and aerodynamics have expanded. In connection with this, one of the contemporary research areas in fluid mechanics and aerodynamics is time series forecasting. Some of the studies are briefly presented in the literature review section. However, it is obvious that it will be an important advantage in engineering to make accurate forecasting especially in these areas with their high practical value. For example, consistent estimates for periods determined in a wind energy system will provide optimum options for increasing the efficiency of the system. If we consider aircraft engines as another example, it is not difficult to say that an accurate future predicted algorithm will reduce the cost on these issues when we consider how much it costs to do fatigue analysis of some materials. In gust loads, which is the subject of this study, ANN time series forecasting studies will be very useful in practical terms. Gust loads on aircraft may induce detrimental influences such as increased aerodynamic and structural loads, structural deformation and decreased flight dynamic performance [1]. The ability to describe accurately the gust load characteristics in which aircraft operates is of importance to the design of new aircraft as well as to the prediction and tracking of damage on existing aircraft. As a result, successful prediction models in aerodynamics could be useful in development of analysis of structural loads and analysis of flight dynamics.

ANN connect inputs and outputs through mathematical models, by training with data given. Thus, it can learn the mechanism and produce outputs for inputs other than the existing ones [2]. Recurrent Neural Network (RNN) is a type of ANN that is

specialized in sequential data. And Long Short-Term Memory (LSTM) is the sub-branch of the RNN that can perform better in time series forecasting [3].

In this study, first of all, literature research is reviewed. Then ,experimental study is presented. It should be noted that the datasets used in this study are retrieved from Flat Plate Wing Transient Gust Experiment performed in Istanbul Technical University Trisonic Research Laboratory by Prof. Dr. Okşan Çetiner Yıldırım and her crew. This experimental study contains a wing model that at some time interval be exposed to a sudden gust. Then LSTM model was created and choosing of parameters of the model was examined. To have a better model in terms of this unique dataset, tuning was performed. Time series predictions were made for different conditions and capability of the system was discussed to create better models in the future.

1.1 Purpose of Thesis

The purpose of this study is predicting the aerodynamic force exerted on an wing model in a flow which the flow includes transient gust, with the help of a LSTM model, and examine the accuracy of the model by changing the parameters of the model and by presenting different configurations.

Therefore, it is aimed to investigate the usefulness of LSTMs forecasting capability in aerodynamic problems.

1.2 Literature Review

In this section, some of the inspiring research papers about time series forecasting with neural networks in fluid mechanics/ aerodynamics are examining.

ElSaid et al. investigated the prediction of aircraft engines' vibrations with a LSTM model. In this paper, train dataset taken from the airlines that experienced excessive engine vibrations in the past is used. This large dataset is characterized by the parameters, which are highly associated with the engine vibrations like angle of attack (AoA), bleed pressure and turbine inlet temperature. The procedure is by using three different LSTM architecture, predicting the next 5s, 10s, and 20s intervals. There are 41431 data from 28 flights used for training, and 38126 data from 57 flights used for testing. As a result, there is 3.3% mean absolute error for 5s

forecasting, 5.51% mean absolute error for 10s forecasting, and 10.19% mean absolute error for 20s forecasting. The paper claims that LSTM is a suitable tool for predicting the aircraft engines' vibration and it can be used in warning systems that are necessary for possible vibration problems [4].

Kulkarni, Dhoble and Padole examined whether the fatigue analysis of a wind turbine blade can be performed by estimating long-term wind parameters using LSTM and nonlinear autoregressive neural networks. In this paper, the blade was created by a finite element model and then analysis of natural frequency was obtained. As a result, paper concludes that LSTM is better in terms of long-term analysis compared to nonlinear autoregressive neural networks [5].

Shi et al. studied on flight trajectory prediction based on a LSTM algorithm. The dataset comes from Automatic Dependent Surveillance Broadcast for once in every 15 seconds between the five months period from June 2017 to November 2017. 95% of this data used for training and remaining part used in testing. As input parameters, position, speed and heading were used. LSTM network was built on Keras library. To maintain continuity, sliding window with 10 values intervals were applied. As a result, time stamp, latitude, longitude and altitude predictions were made and these predictions were analyzed by the mean absolute error, the mean root error, the root mean squared error and the discrete wavelet transform methods. This paper emphasizes that this LSTM predictions outperforms the other prediction methods and presents a strong basis for abnormal detection and decision-making in Air Traffic Management [6].

Su, Yu and Tan studied on a LSTM based wind power forecasting method. For the purpose of getting more accurate ultra-short-term wind power forecasting results, their method took wind turbine states and wind speed components into account. Specifically, use of wind turbine states such as yaw error and rotor speed developed the efficiency of prediction model because this approach was similar to the actual situtation. Considering yaw error and rotor speed of wind turbine states, the mean absolute percentage error at the 5s time interval was measured as 2.72%. This work showed that wind speed components and wind turbines states can be used as the network input in wind power forecasting [7].

2. EXPERIMENTAL DATA

The datasets was taken from the experimental studies which were performed by Prof. Dr. Okşan Çetiner Yıldırım. Let's briefly explain this experiment: There is a wing model which has the dimensions of chord length (c) 100 mm, span (s) 200 mm, and thickness (t) 5 mm. There is also a gust plate, which is fixed centrally to a certain distance from the wing model in the x-direction. Gust plate has the dimensions of chord length (c) 100 mm, span (s) 400 mm, and thickness (t) 5 mm. In y-direction, there is a distance between the gust plate center and the wing chord line at 0 AoA which is called as Y-Shift parameter. A flow of 0.1 m/s in x-direction is applied for 60s. Between 5s and 9s of the flow, gust plate makes a move of turn over. Experiment is illustrated in **figure 1**.

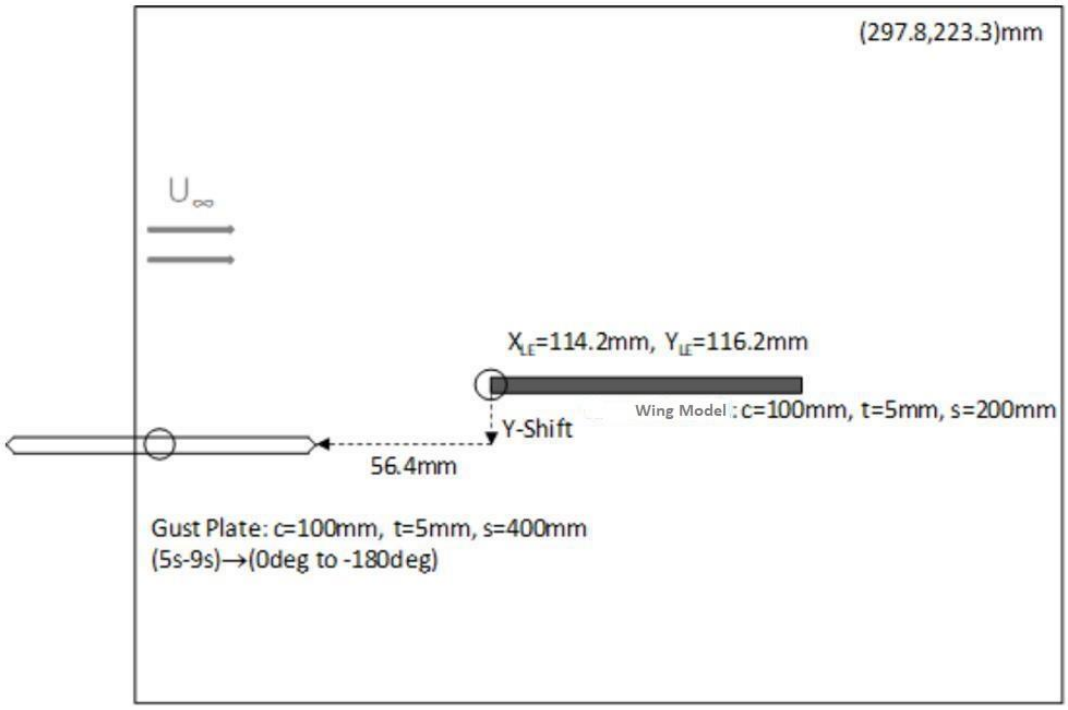

Figure 1: Illustration of the experiment.

Y-Shift and AoA of the wing model are changing parameters of the experiment. For Y-Shift, three configurations were used which are 19 mm, 29 mm, and 39 mm. On the other hand, 7 different AoA values ($0^{\circ}, 5^{\circ}, 10^{\circ}, 15^{\circ}, 20^{\circ}, 30^{\circ}, 45^{\circ}$) were set for each of these Y-Shift values. In total, 21 different configurations were used. 5 different datasets of 60 seconds (sometimes 60.01) were obtained for each of these 21 configurations to extract the effect of noise. Measurement frequency of the experiment is 1000 Hz. In 60 seconds 60,000 values (sometimes 60100 in 60.01 seconds) of normal force, axial force, side force, yaw moment, roll moment and pitch moment were obtained. In **figure 2**, example of an Fx distribution is seen. Gust effect of 5-9 s interval is obvious. In **figure 3**, same graph is plotted for Fy.

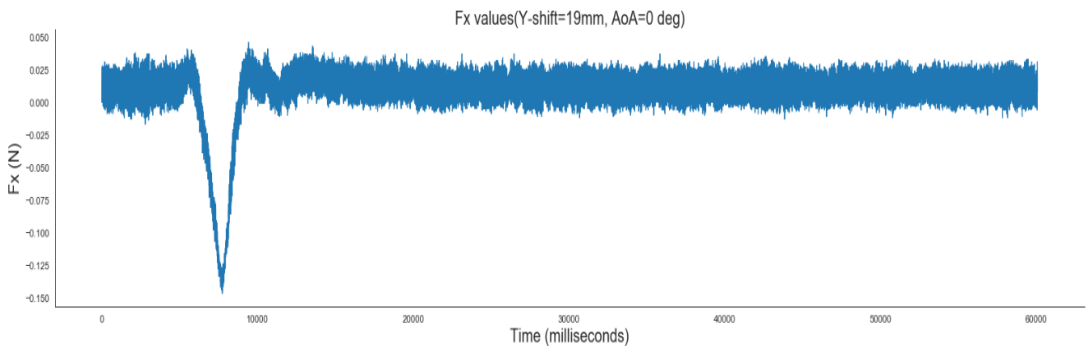

Figure 2: Fx Distribution

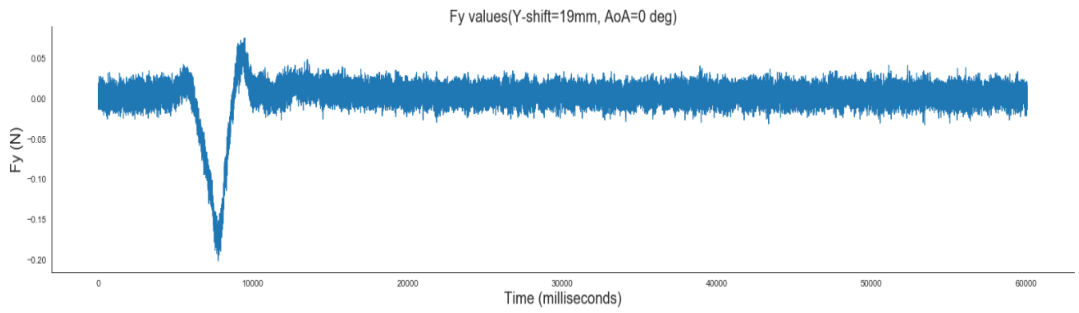

Figure 3: Fy Distribution

3. ARTIFICIAL NEURAL NETWORKS

Artificial Neural Network is a type of machine learning algorithm that built on inspiration of the biological neural systems. Just like the biological systems, ANNs are strong in terms of information procesessing abilities. These abilities can be summarized as learning nonlinearity easily, adapting correlation, capability of generalizing and to be able to figure out abstract information. Other similarity is that, the learning process of the ANNs are also highly related with the experience, just like in humans, which corresponds to the training of an ANN. That is why, ANNs are popular computation tools to solve complex problems [1].

Building block of an ANN is called as node, or neuron, or perceptron. Each neuron takes input and use activation function and produce an output. Neurons are pieced together into layers. There is a layer of input neurons, a layer of output neurons, and between these two, there is one or more hidden layers. The connections between these layers are called as networks. Every connection network has a weight, which shows the significance of the connection [1].

3.1 Recurrent Neural Networks

Recurrent Neural Networks (RNN) are a type of ANN that use previous outputs as inputs together with hidden layers. In a classical ANN, all the inputs and outputs are independent of each other. On the other hand, RNNs are built on the idea of using sequential information. They are called “recurrent” because they have memory about the past information. Because of that, RNNs’ most important feature is hidden layers. Theoratically RNNs are capable of using all the past information, however in practice use of past information is limited [8]. RNNs are widely used in language modelling, text generating, machine translation, speech recognition, music generation, and time series forecasting. A typical scheme of an RNN is shown in **figure 4**.

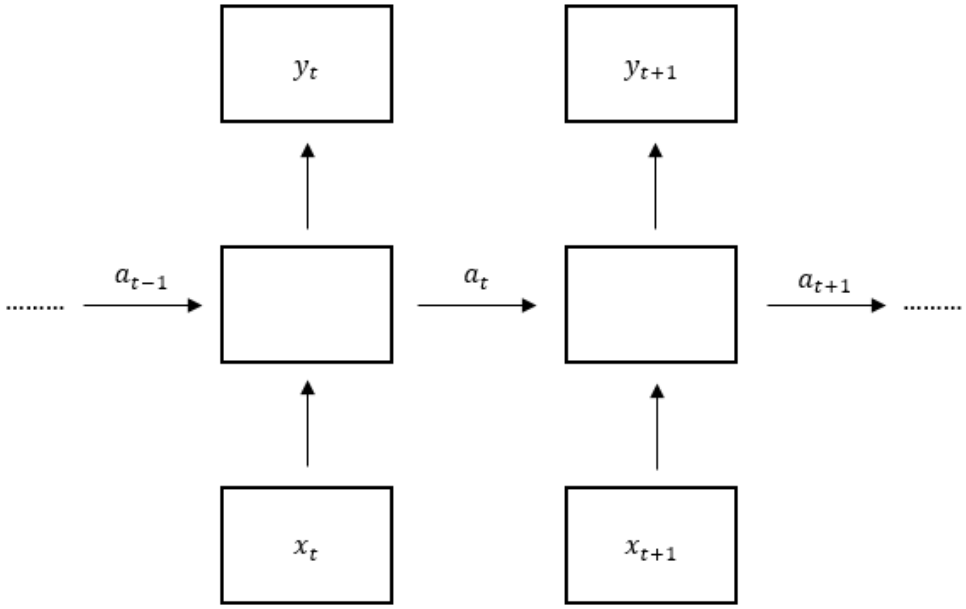

Figure 4: Schematic of a RNN [9]

For each time step t, the activation a_t and the output y_t can be expressed as follows:

$$a_t = g_1(W_{aa}a_{t-1} + W_{ax}x_t + b_a) \quad \textbf{(3.1)}$$

And

$$y_t = g_2(W_{ya}a_t + b_y) \quad \textbf{(3.2)}$$

Here W_{ax}, W_{aa}, W_{ya}, b_a, b_y are temporarly shaped coefficients. g_1 and g_2 are activation functions. These functions are generally nonlinear such as sigmoid, tanh and ReLU.

One of the biggest problems of RNNs is ”vanishing gradients” problem. This occurs when the information about the input or gradient passes thorough many layers, it will vanish and wash out by the time when it reaches to the end or beginning layer. Because of this problem of RNNs, in time series forecasting which requires the long-term dependencies, a special type of RNNs, LSTM Networks are better options [8].

3.2 Long Short-Term Memory Networks

It was mentioned before that the RNNs have difficulties in prediction of sequential time series problems because of the vanishing gradient problem. LSTM is a special type of RNN that can overcome vanishing gradient problem of RNN. Hochreiter and Schmidhuber [10] first introduced LSTM networks. LSTMs are superior in terms of keeping and learning long-term dependencies of inputs compared to RNNs. This superior memory has the ability of remembering information over a long period of time and therefore enables reading, writing, and deleting information from their memories. The LSTM memory is called a gated cell. A LSTM model takes all necessary features from inputs and can keep them for long time. Based on assigned weights in the training process, the decision of keeping or deleting the information is made. Therefore, LSTM can learn which information worth to preserve or delete [8].

A LSTM cell consists of the input gate, the output gate and the forget gate. The input gate selects the new information to add, the forget gate decides to keep or delete the information, the output gate controls the information letting through to next cell [11]. In **figure 5**, each round rectangle is a cell [7].

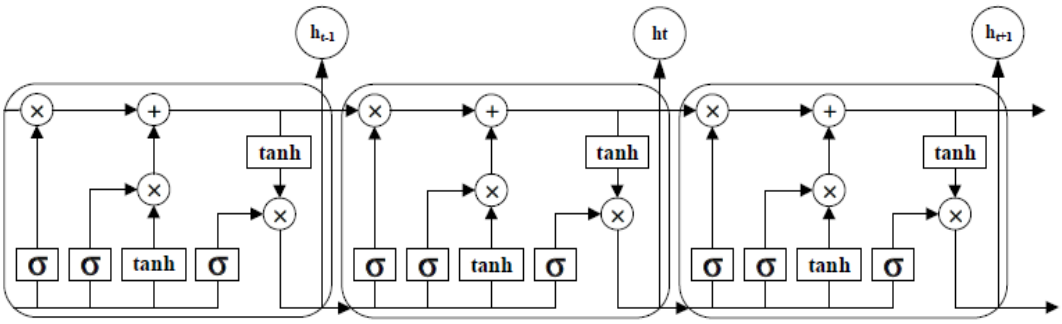

Figure 5: LSTM Structure [7]

As it seen each cell has 4 layers of neural network. The first layer's neurons are sigmoid control layer of the forget gate [7]. In this gate it is decided which information needs to be removed from the memory. This decision is made according to h_{t-1} and x_t. The output of this gate is f_t which takes the value between 0 and 1 where 0 indicates totally forgetting the learned value, and 1 indicates keeping the whole [8]. Forget gate has a critical role in reducing over-fitting by not keeping all

information of the previous time steps [11]. Output is represented by the equation **(3.3)**.

$$f_t = \sigma\ (W_f\ [h_{t-1}, x_t] +\ b_f) \tag{3.3}$$

W_f is the weight of the forget gate and b_f is the bias of the forget gate. In input gate, there are two layers that are sigmoid control layer and tanh layer. Sigmoid layer determines which values are to be updated. Tanh layer creates a vector of new candidate that may be added to the memory. Outputs of these layers computed by the equations **(3.4)** and **(3.5)**, respectively [8].

$$i_t = \sigma\ (W_i\ [h_{t-1}, x_t] +\ b_i) \tag{3.4}$$

$$c_t^{\sim} = tanh(W_c\ [h_{t-1}, x_t] +\ b_c) \tag{3.5}$$

Where W_i is the weight of the input gate, b_i is the bias of the input gate, W_c is the weight of the upgraded value and b_c is the bias of the upgraded value. By combining these two equations, equation **(3.6)** shows the update of the old state of the cell c_{t-1} to the c_t [7].

$$c_t = f_t * c_{t-1} + i_t * c_t^{\sim} \tag{3.6}$$

Finally, the output gate first decides what part of the LSTM memory contributes to the output by using a sigmoid layer. Then makes output value between -1 and 1 with tanh layer. At the end the output of the tanh layer is multiplied by the output of the sigmoid layer. In this way forgetting and memory parameters can be used in output [7]. Equations **(3.7)** and **(3.8)** shows the relations to calculate the output.

$$o_t =\ \sigma\ (W_o\ [h_{t-1}, x_t] +\ b_o) \tag{3.7}$$

$$h_t = o_t * \tanh(c_t) \tag{3.8}$$

Where W_o is the weight of the output gate, b_o is the bias of the output gate, o_t is the output value and h_t is it's representation as between -1 and 1.

4. MODEL

4.1 Statistical Analysis

In this study, mean absolute percentage error (Mean APE), maximum absolute percentage error (Max. APE), mean squared error (MSE) and r-squared (R^2) are used as statistical analysis approach.

MSE was used in training process of the LSTM model as loss function. It is shown in equation **(4.1)**.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Actual - Predicted)^2 \tag{4.1}$$

Mean APE, Max. APE and (R^2) were used in the analysis of the results. Note that, in the calculation of APE, global range was used. Equations are represented as follows.

$$Mean\ APE = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{|Actual - Predicted\ |}{|Actual_{max} - Actual_{min}|}x100\right) \tag{4.2}$$

$$Max.APE = \max\left(\frac{|Actual - Predicted\ |}{|Actual_{max} - Actual_{min}|}x100\right) \tag{4.3}$$

$$R^2 = 1 - \frac{Prediction\ Variation}{Actual\ Variation} \tag{4.4}$$

4.2 Keras

In this study, Keras neural network Application Programming Interface (API) is used. Keras is a deep learning API written in Python, running on top of the machine learning platform

TensorFlow [12]. Keras has ben succesful in terms of sequential models. It is also easy to use and provides fast results.

4.3 LSTM Model

LSTM models involve many parameters that effect the performance of the model. Therefore, it is important to choose proper parameters. Most of the parameters were chosen based on the literature research or default options. These are dropout, loss function, optimizer, activation functions, number of epochs, batch size, and early stopping function. For some critical parameters, tuning is necessary. There are different optimization techniques; however, these techniques' success profoundly depends on the characteristics of the data. Application of these complicated techniques might be another research topic. In this model, tuning was made according to three parameters, which are number of training data, number of lookback data and number of neurons in a single layer.

Dropout is a technique to improve neural networks by means of reducing overfitting. It regularizes the LSTM by preventing some of the inputs and recurrent connections update weight and activation. This method is very usefull especially in networks which are trained on small datasets because small datasets are highly sensitive to overfitting [13]. In this study, dropout is not used because the size of the data is large enough that overfitting is not a problem to deal with.

There should be a loss function for training part of the model. Loss functions are used to determine the loss of the model and update the weights, so that to make loss smaller. If training loss is smaller than the validation loss then the model is overfitting, if training loss higher than the validation loss, then model is underfitting. Desired is that the training loss converge on the validation loss and be small. In this LSTM model, as a loss function Mean Squared Error (MSE) is used. MSE is the simplest and most common loss function in artificial intelligence. MSE is calculated as the average of the squared differences between the predicted and actual values. The squaring means that larger mistakes result in more error than smaller mistakes, meaning that the model is punished for making larger mistakes [3].

Optimizers are important to increase the accuracy of the model. For the optimizer, adaptive moment estimation (Adam) is chosen. Adam is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments [14]. It has been very succesful in most practical applications.

Activation functions determine the output of a neural network. The function is attached to each neuron in the network, and determines whether it should be activated ("fired") or not, based on whether each neuron's input is relevant for the model's prediction. In Keras, sigmoid is the default recurrent activation function and the hyperbolic tangent (tanh) is default output activation function. In this model these two are used as activation functions [15].

Number of epochs is the number of times to expose the model to the whole training dataset. Batch size is the number of samples within an epoch after which the weights are updated. The size of a batch must be more than or equal to one and less than or equal to the number of samples in the training dataset [3]. In general, large batch sizes result in quick training, but may not always converge equally quickly. The little batch sizes train very slowly, but converges fast. Also in general, models give better results when there is more epochs in training. However, these parameters are not easy to determine. General approach is determine them with trying different values and seeing which numbers work better. In this model number of epochs is used as 500 and batch size is determined as 200.

Early stopping parameter works to prevent model overfitting during training. Since the main purpose of the training is minimazing the loss, it is not desired that loss becoming higher after some point. In this model, early stopping mechanism use validation loss as monitor and check at end of every epoch whether the loss is no longer decreasing. If the loss is no longer decreasing then training stops. Another useful argument in early stopping parameter is "patience" which shows the number of epochs with no improvement after which training will be stopped [16]. Patience set as 20.

4.3.1 Tuning The LSTM Model

Number of training data is very important parameter for all the artificial intelligence applications. It is highly depended on the data structure so it should be determined by trying. In our experimental data, there is a gust effect between 5 seconds and 9 seconds. Since one of the purposes of forecasting is validating the effect of upcoming gust, training should be made in the first 5 seconds data. Therefore, to determine the proper number of training data, seven discrete datasets were used. These sets are from the different angle of attacks ($0^\circ, 5^\circ, 10^\circ, 15^\circ, 20^\circ, 30^\circ, 45^\circ$) in 19 mm y-shifted situation. As candidates of number of training data, 156, 312, 625, 1250, 2500 and 5000 were chosen. Model was run for all these training sets for seven angle of attacks. Average of these forecastings' absolute mean percentage error, absolute maximum error and R-squared measurements are seen in **table 1** below. Besides 156,

all others gave acceptable results. Since average of the maximum absolute percentage error (max APE) is minimum for 625 and, average of the mean absolute percentage error (mean APE) is minimum for 5000, these two were chosen as number of training data.

Table 4.1 : Results according to number of training data.

Number of Training Data	% Absolute Mean Error (average)	% Absolute Maximum Error (average)	R^2 (average)
156	20.3522	34.0659	-2.7634
312	1.2532	8.7081	0.9775
625	1.2135	7.7182	0.9795
1250	1.0982	8.0969	0.9831
2500	1.0491	9.3124	0.9831
5000	0.9998	9.5811	0.9849

LSTM network uses past information in time series forecasting. Look back is the number of lag observations to use as input to the model. It can be called as window too. Since it greatly affects the performance of the model, good choice is necessary. To determine look back parameter, similar procedure to number of training data choosing procedure was applied. Seven discrete datasets from the different angle of attacks in 19 mm y-shifted situation were used. As candidates of look back range, 15, 30, 60 and 120 were chosen. Average of these forecastings' mean APE, max APE and R-squared measurements are seen in **table 2** below. For both 625 and 5000 training data, 30 look back gave the best results.

In this model, single hidden layer was used. Although there is no strict rule on how many neurons should chosen, a trial and error approach can provide the best option for this model. Same procedure was applied and results are listed in **table 3**. All the options seem reasonable. However, 50, 100 and 200 gave slightly better results. Number of neurons (nodes) in single layer was determined as 100.

Table 2 : Results according to look back parameter

Number of Training Data	Look back	% Absolute Mean Error (average)	% Absolute Maximum Error (average)	R^2 (average)
625	15	5.6192	17.2617	0.6440
	30	1.2135	7.7182	0.9795
	60	1.3130	10.9908	0.9743
	120	1.4445	11.8109	0.9693
5000	15	1.2570	16.7423	0.9628
	30	0.9998	9.5811	0.9849
	60	1.1947	10.5364	0.9786
	120	1.3926	13.9114	0.9684

Table 3 : Results according to number of neurons

Number of Training Data	Number of Neurons	% Absolute Mean Error (average)	% Absolute Maximum Error (average)	R^2 (average)
625	25	1.4962	13.1350	0.9650
	50	1.2764	8.5971	0.9767
	100	1.2135	7.7182	0.9795
	200	1.1748	7.9572	0.9807
	400	4.0567	14.2855	0.7990
5000	25	1.0590	9.7426	0.9825
	50	1.0338	9.6595	0.9832
	100	0.9998	9.5811	0.9849
	200	1.0147	10.1535	0.9839
	400	1.0146	10.1822	0.9837

4.4 Moving Average

The moving average is a simple technical analysis tool that smooths out data by creating a constantly updated average value. The average is taken over a specific period of time, like 10, 20 or 100. For example, a plot of a moving average of 20 observations simply displays the average of the twenty most recent observations. Advantage of moving average method is that it smoothize the data and hide noises. However, it should be applied carefully because it can reflect the character of the dataset wrongly when the period is selected high [3]. In this study, the LSTM model was applied using the moving average method and the results were compared with the main model. It should be noted that 20 is selected as the period in moving average model.

5. RESULTS

Results of the model are shown in this section. The predictions were compared with the actual data. In the analysis of comparisons, two statistical methods were used: mean APE and max APE.

The result analysis was made on four different parameters. In the estimation of the force in the x-direction, the y-shift effect was examined. With this, the impact of the training dataset has also been demonstrated. As another comparison parameter, the result of the prediction using the moving average method for the force in the x direction was examined. As the last parameter, prediction of the force in the y direction was shown.

Figure 6 and **Figure 7** show that in the case y-shift being 19 mm, the training dataset being 625 or 5000 did not determine the result. This is also valid for the change in AoA. The results obtained by the model did not change as AoA changed. In all these cases, mean APE is around 1%, and max APE is around 10%. When we look at the average error, it can be said that the model worked successfully.

This interpretation is also valid in cases where y-shift is 29 mm and 39 mm. **Figure 8**, **Figure 9**, **Figure 10** and **Figure 11** show that the training data set being 625 or 5000 has no decisive effect. As AoA changes, the percentage of error does not change significantly. Most importantly, the effect of y-shift change on the performance of the model was not observed. As it is seen, while y-shift is 29 mm and 39 mm, the mean APE and the max APE are still in 1% and 10% bands, respectively.

Figure 12 and **Figure 13** show the comparison of the cases, where the moving average method applied model and the normal model. For the analysis, y-shift was chosen as 19 mm and training dataset as 5000. It can be seen in the results that although the moving average applied model did not make a big difference in percentage, it is characteristically more successful. The probable reason for this may be that the noise in the dataset is partially removed by moving average.

Finally, the success of the model's predictions of the force in the y direction was compared to the success of the predictions in the x direction. As can be seen in the **figure 14** and **figure 15**,

the model slightly worse in the y-direction force prediction. This may be related with the gust's character creating more turbulence in the y direction.

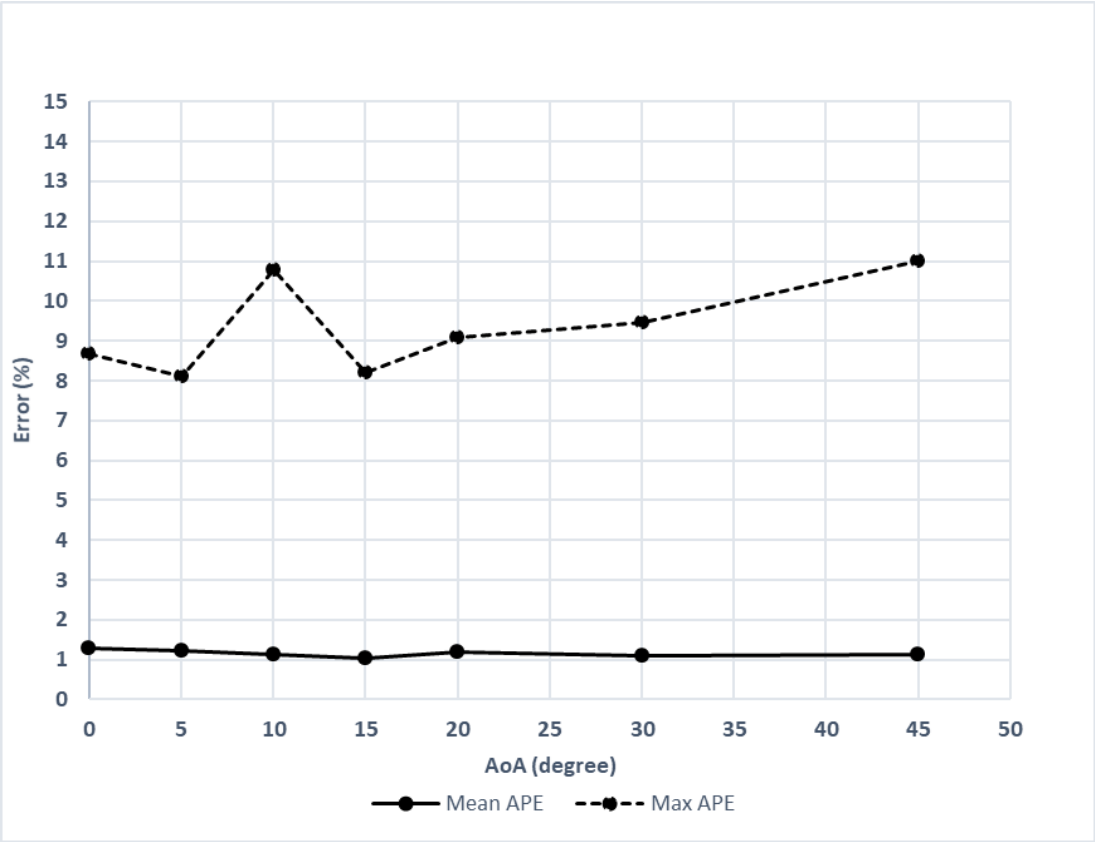

Figure 6: Percentage errors of the prediction (case: Y-Shift= 19 mm and Training Data= 625)

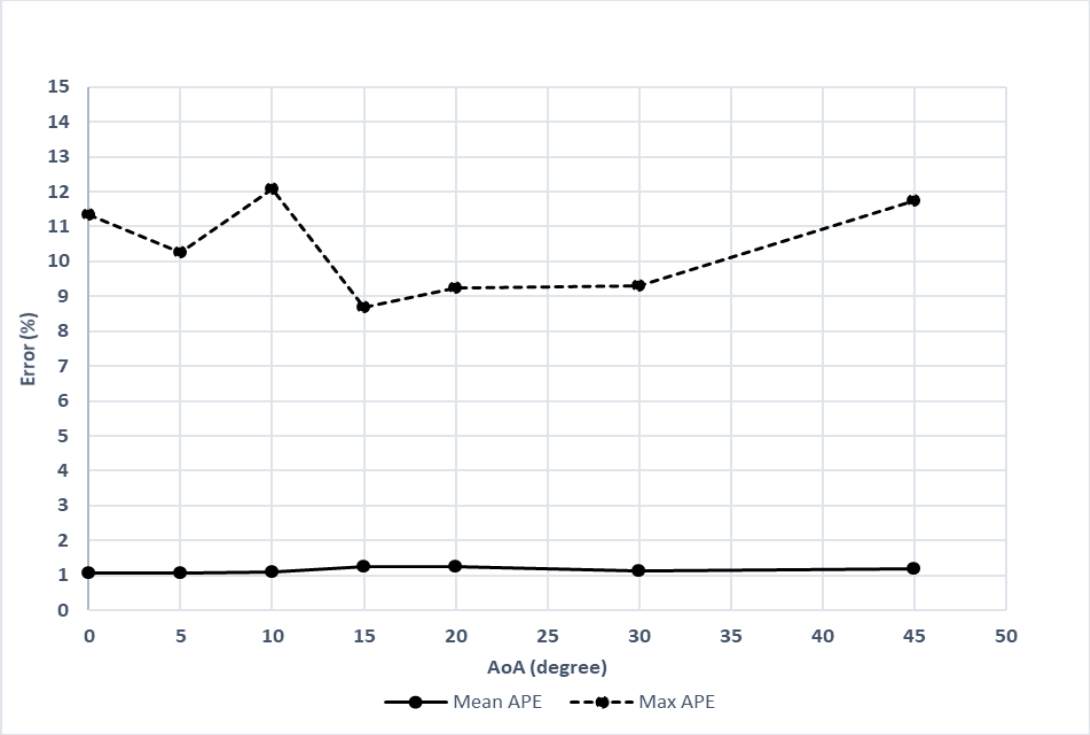

Figure 7: Percentage errors of the prediction (case: Y-Shift= 19 mm and Training Data= 5000)

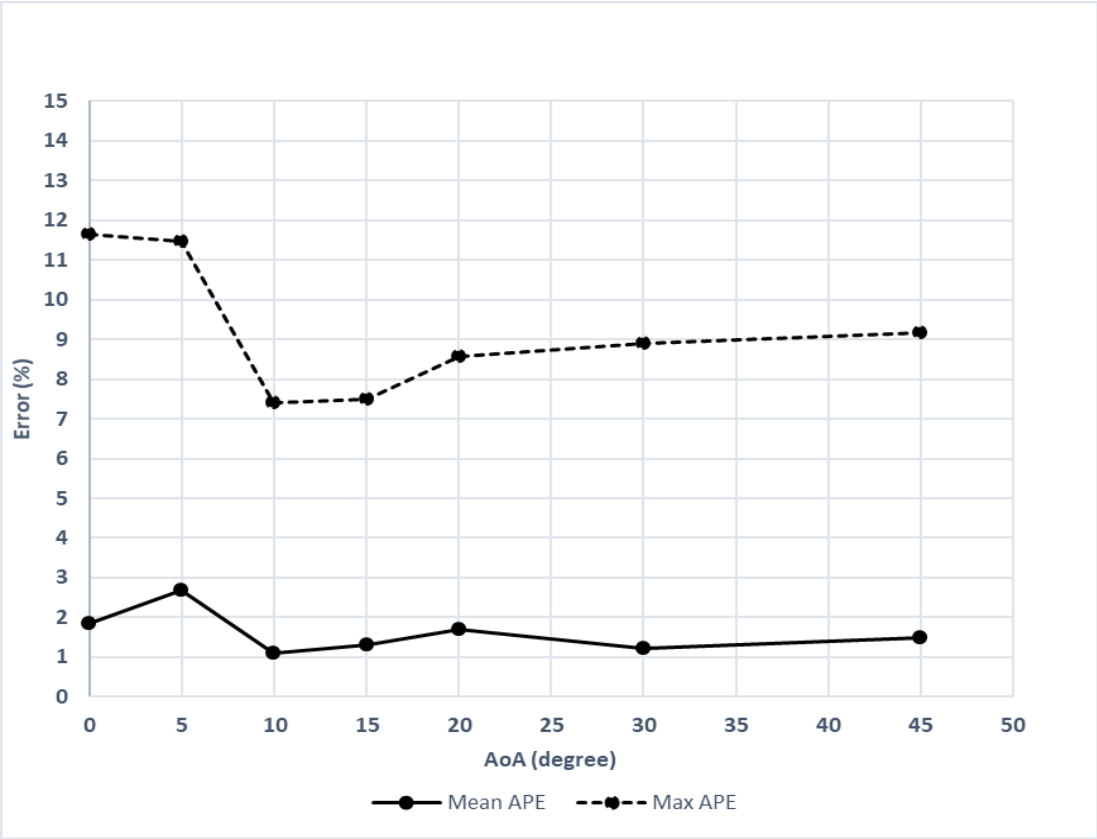

Figure 8: Percentage errors of the prediction (case: Y-Shift= 29 mm and Training Data= 625)

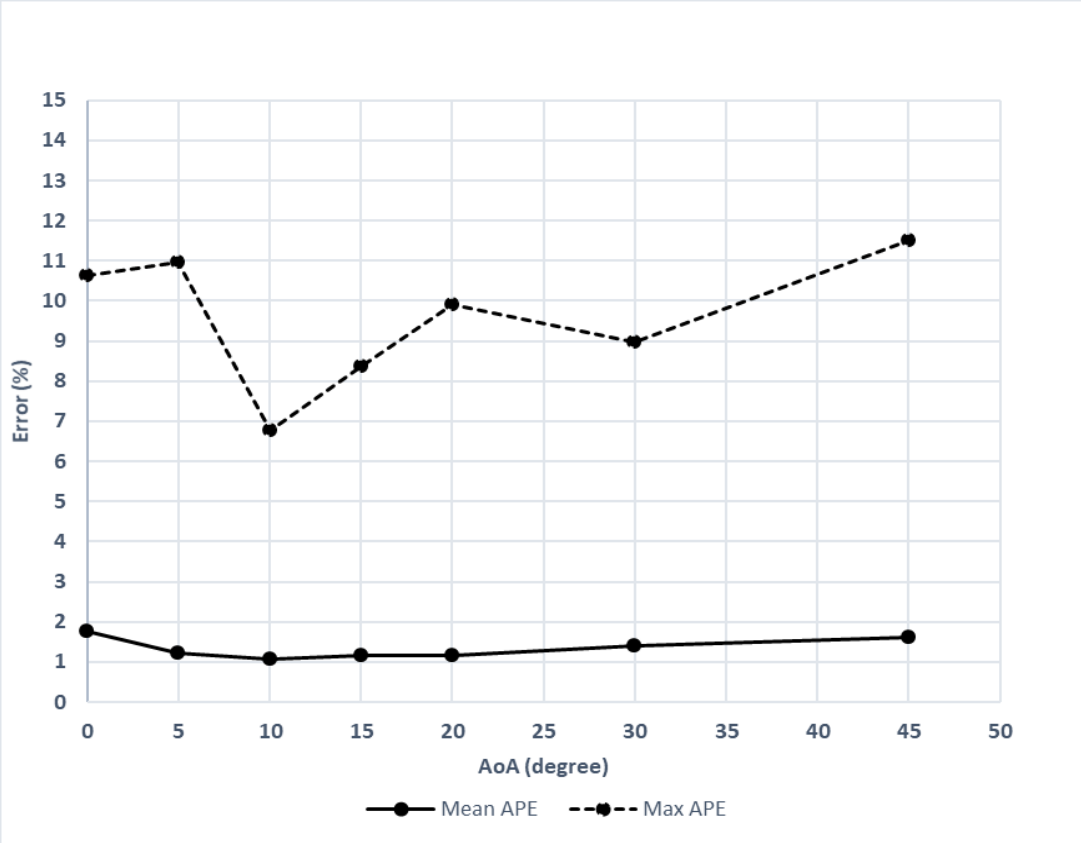

Figure 9: Percentage errors of the prediction (case: Y-Shift= 29 mm and Training Data= 5000)

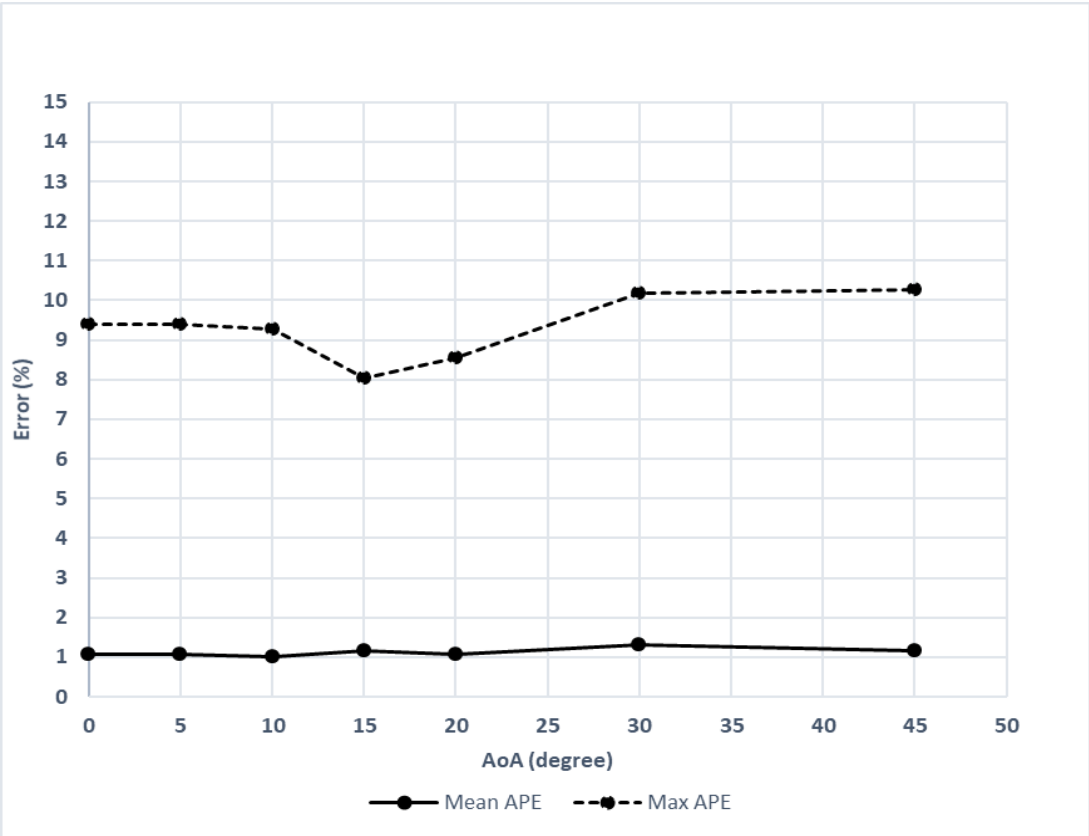

Figure 10: Percentage errors of the prediction (case: Y-Shift= 39 mm and Training Data= 625)

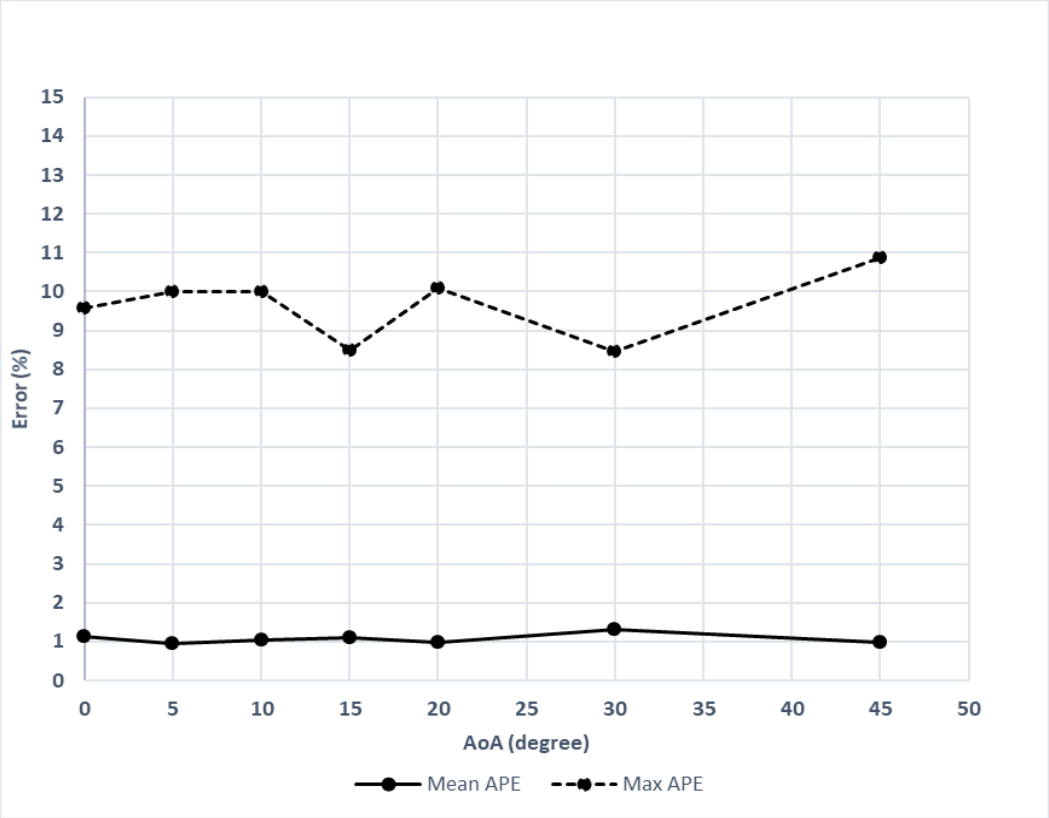

Figure 11: Percentage errors of the prediction (case: Y-Shift:39 mm and Training Data:5000)

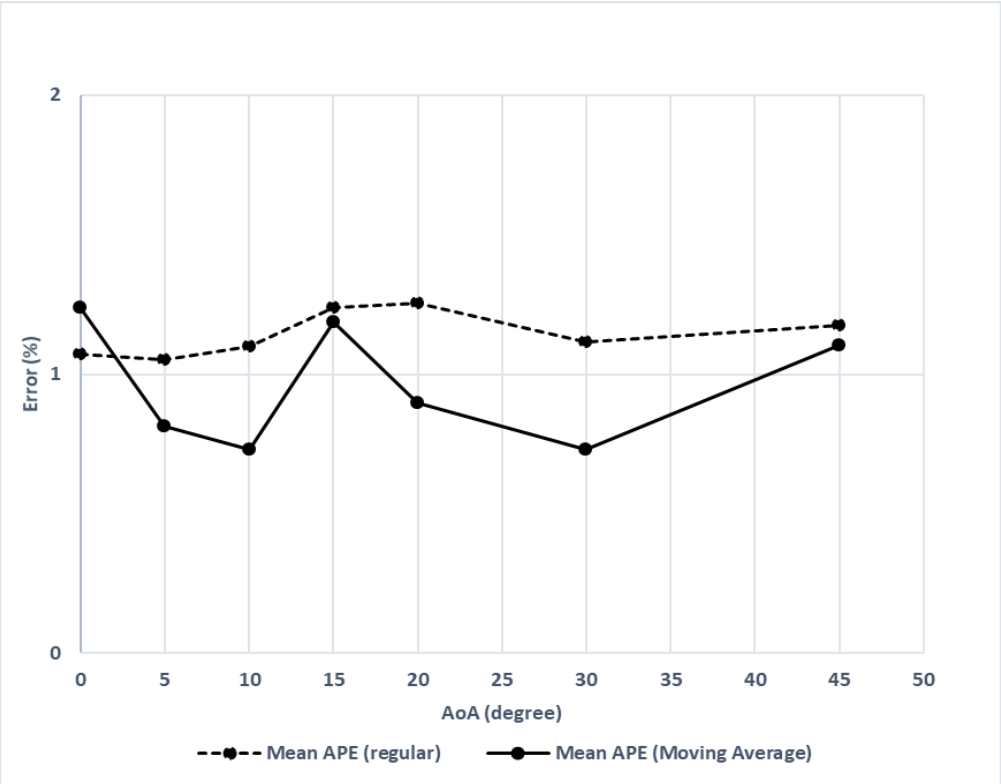

Figure 12: Comparison of Mean APEs: the regular model and the moving average applied model (case: Y-Shift=19 mm, Training Data: 5000)

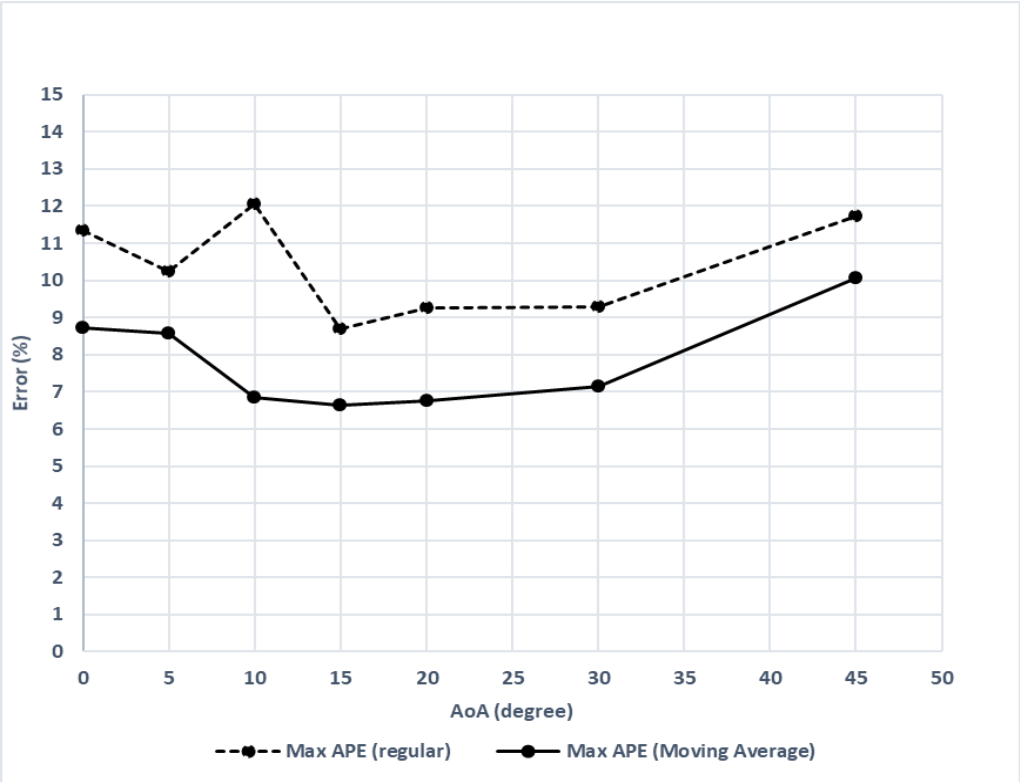

Figure 13: Comparison of Max APEs: the regular model and the moving average applied model (case: Y-Shift=19 mm, Training Data: 5000)

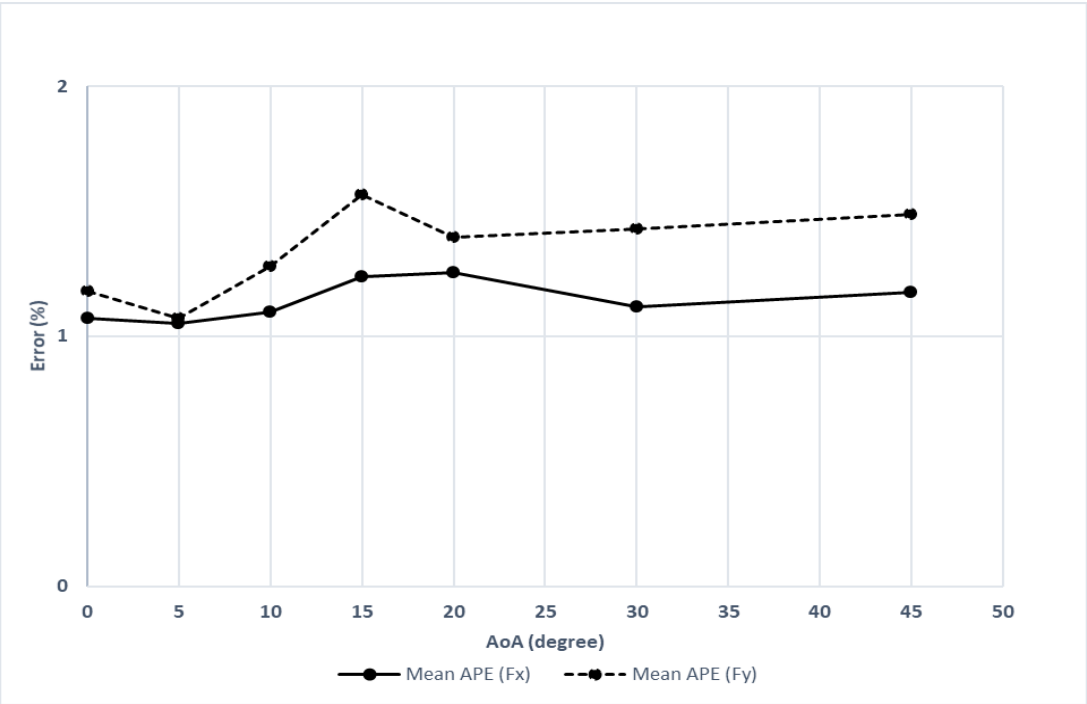

Figure 14: Comparison of Mean APEs: Fx forecasting and Fy forecasting (case: Y-Shift=19 mm, Training Data: 5000)

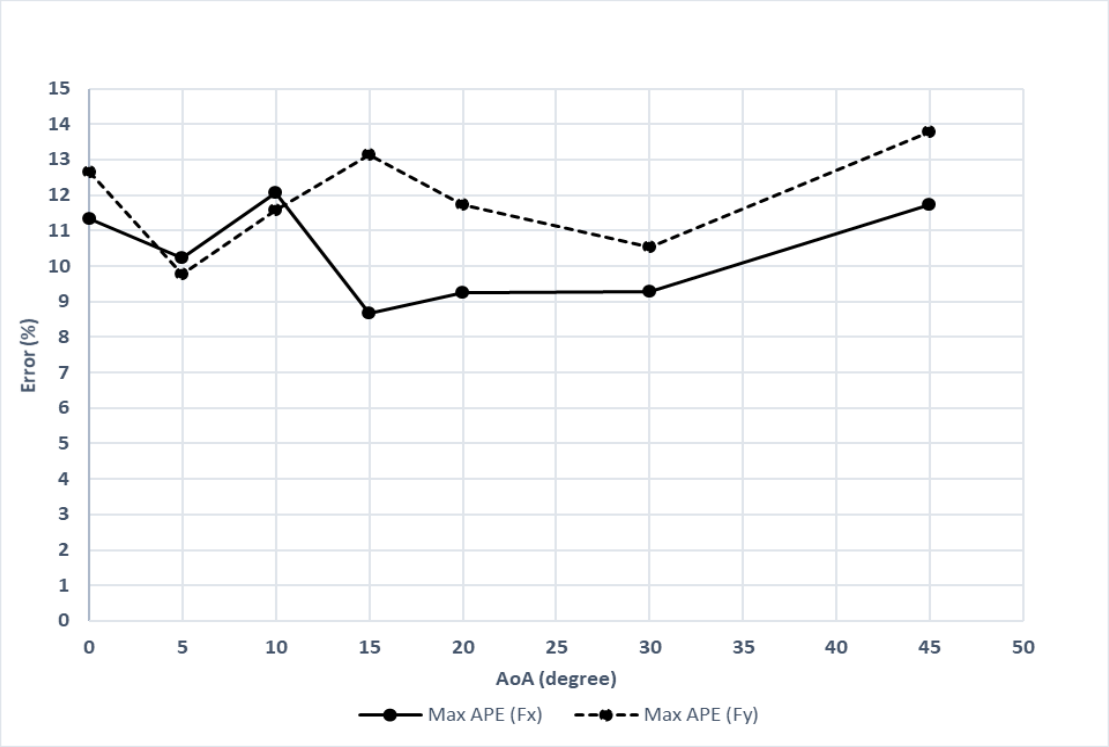

Figure 15: Comparison of Max APEs: Fx forecasting and Fy forecasting (case: Y-Shift=19 mm, Training Data: 5000)

6. COMMENTS

ANN has an algorithm structure that enables connection between many complex and nonlinear numerical data. LSTM models are one of the most useful architectures in ANNs in order to make successful predictions by making use of historical data. In terms of aerodynamics and fluid mechanics, making predictions using LSTM can be very useful. In this study, an LSTM model has been established based on experimental data examining the transient gust effect on a wing model. In this LSTM model, the number of training data, look back parameter and number of neurons in a single hidden layer were determined by tuning. The rest of the parameters were selected from the default options or by scanning the literature, among the options suitable for this particular data.

As a result, LSTM model made predictions based on the force in the x direction, the number of training data, the y-shift parameter, the moving average method and the force in the y-direction. These predictions have generally been successful. In comparisons among themselves, it was revealed that the estimation of the force in the x direction was more successful because the gust had a more turbulent effect in the y direction. In addition, the moving average method was predicted with more satisfactory error rates than the normal method.

This graduation project study has shown that the use of artificial neural networks in aerodynamics can be applied in many creative subjects and studies that can save time and cost in terms of engineering. In future studies, methods that can provide optimization of all parameters of LSTM model can be studied. On the other hand, moment parameters can be added to the training dataset and analyzed to evaluate the success of the model.

REFERENCES

[1] **Wu, Z., Cao, Y. and Ismail, M.**, 2019. Gust loads on aircraft. *The Aeronautical Journal*, 123(1266), pp.1216-1274.

[2] **Basheer, I. and Hajmeer, M.,** 2000. Artificial neural networks: fundamentals, computing, design, and application. *Journal of Microbiological Methods*, 43(1), pp.3-31.

[3] **Brownlee, J.**, 2018. *Deep Learning For Time Series Forecasting: Predict The Future With Mlps, Cnns And Lstms In Python.*

[4] **ElSaid, A., Wild, B., Higgins, J. and Desell, T.,** 2016. Using LSTM recurrent neural networks to predict excess vibration events in aircraft engines. *2016 IEEE 12th International Conference on e-Science* (e-Science).

[5] **Kulkarni, P., Dhoble, A. and Padole, P.**, 2018. Deep neural network-based wind speed forecasting and fatigue analysis of a large composite wind turbine blade. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 233(8), pp.2794-2812.

[6] **Shi, Zhiyuan & Xu, Min & Pan, Quan & Yan, Bing & Zhang, Haimin**. (2018). LSTM-based Flight Trajectory Prediction. 1-8. 10.1109/IJCNN.2018.8489734.

[7] **Y. Su, J. Yu, M. Tan, Z. Wu, Z. Xiao and J. Hu**, "A LSTM Based Wind Power Forecasting Method Considering Wind Frequency Components and the Wind Turbine States," *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, Harbin, China, 2019, pp. 1-6, doi: 10.1109/ICEMS.2019.8921671.

[8] **S. Siami-Namini, N. Tavakoli and A. S. Namin,** "The Performance of LSTM and BiLSTM in Forecasting Time Series," *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019, pp. 3285-3292, doi: 10.1109/BigData47090.2019.9005997.

[9] **Zhou, S., Rueckert, D. and Fichtinger, G.**, n.d. *Handbook Of Medical Image Computing And Computer Assisted Intervention*.

[10] **Hochreiter, Sepp & Schmidhuber, Jürgen.** (1997). Long Short-term Memory. *Neural computation*. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[11] **Mohan, Arvind & Gaitonde, Datta**. (2018). A Deep Learning based Approach to Reduced Order Modeling for Turbulent Flow Control using LSTM Neural Networks.

[12] **Team, K.**, 2020. *Keras Documentation: About Keras*. [online] Keras.io. Available at: https://keras.io/about/.

[13] **Srivastava, Nitish & Hinton, Geoffrey & Krizhevsky, Alex & Sutskever, Ilya & Salakhutdinov, Ruslan**. (2014). Dropout: A Simple Way to Prevent Neural

Networks from Overfitting. *Journal of Machine Learning Research*. 15. 1929-1958.

[14] **Team, K.**, 2020. *Keras Documentation: Adam*. [online] Keras.io. Available at: https://keras.io/api/optimizers/adam/.

[15] **Team, K.**, 2020. *Keras Documentation: Layer Activation Functions*. [online] Keras.io. Available at: https://keras.io/api/layers/activations/.

[16] **Team, K.**, 2020. *Keras Documentation: Earlystopping*. [online] Keras.io. Available at: https://keras.io/api/callbacks/early_stopping/.