ESTIMATING INSTRUMENTATION DATA ACQUIRED DURING FLIGHT TEST FROM A HELICOPTER ENGINE USING PREDICTOR MODELS

HELIKOPTER MOTORUNDAN TOPLANAN UÇUŞ TEST ENSTRUMANTASYON VERİLERİNİN ÖNGÖRÜCÜ MODELLERLE TAHMİN EDİLMESİ

BARIŞ COŞKUN

ASST. PROF. DR. BURKAY GENÇ

Supervisor

Submitted to

Graduate School of Science and Engineering of Hacettepe University

as a Partial Fulfillment to the Requirements

for he Award of Degree of Master of Science

in Computer Engineering

2021

Aileme...

ABSTRACT

ESTIMATING INSTRUMENTATION DATA ACQUIRED DURING FLIGHT TEST FROM A HELICOPTER ENGINE USING PREDICTOR MODELS

Barış Coşkun

M.Sc., Department of Computer Engineering

Supervisor : BURKAY GENÇ

June 2021, 57 pages

Flight test of a helicopter is the most dangerous phase of its development project. Any minor mistake or miscalculation during a test could lead to a catastrophic accident. In order to protect the health of a helicopter during a test, lots of sensors are placed on critical parts and equipment of the helicopter. Data acquired from these sensors are monitored by the experts from different fields through the test and in any kind of unexpected or out of limit data, these experts interfere with the test procedure. Acquiring miscalibrated sensor data or even losing the sensor completely would increase the potential risk and delay the test although there is no problem on helicopter. Also having miscalibrated or no data from a sensor makes a part of the helicopter invisible for analysis. Due to all these reasons, we propose a hybrid system composed of decision tree that calculates the noise ratio of the sensor readings and neural network that replaces miscalibrated ones with accurate predictions based on the other sensor readings. In this paper, we present the models and how we construct them, test them on

semi-synthetic data and show that these models can be used in production and testing systems in real life.

Keywords: Helicopter, Flight Test, Instrumentation, Data Confidence, Data Prediction

ÖZET

HELİKOPTER MOTORUNDAN TOPLANAN UÇUŞ TEST ENSTRÜMANTASYON VERİLERİNİN ÖNGÖRÜCÜ MODELLERLE TAHMİN EDİLMESİ

Barış Coşkun

Yüksek Lisans, Bilgisayar Mühendisliği

Tez Yöneticisi : BURKAY GENÇ

Haziran 2021, 57 sayfa

Helikopter tasarım sürecinin en tehlikeli kısmı uçuş test aşamasıdır. Testler sırasında yapılacak en ufak hata veya yanlış hesaplamalar ciddi kazalara sebep olabilir. Bu sebeple uçuş testleri sırasında helikopterin sağlık durumunun takip edilebilmesi için özellikle kritik bölgeler olmak üzere helikopterin çoğu parçası algılayıcılarla enstrümante edilir. Algılayıcılardan toplanan veriler anlık olarak yer istasyonuna iletilip alanlarında uzman kişiler tarafından yorumlanarak beklenmedik bir durumun önceden fark edilmesi sağlanır. Ancak algılayıcı veya veri toplama ünitelerinde oluşacak bir arıza sonucunda toplanan verinin güvenilirliği kalmaz. Bu gibi durumlarda veri analiz ekipleri tarafından gelen veri yanlış yorumlanarak gereksiz yere testin kesilmesine veya öngörülebilecek bir kazanın fark edilmemesi gibi ciddi sonuçlara yol açabilir. Bu tez kapsamında oluşturulan hibrit sistemde Karar Ağacı modeli kullanılarak verideki bozulma miktarı ve Yapay Sinir Ağı kullanılarak da olması gereken

verinin tahmini gerçekleştirilmektedir. Oluşturulan bu sistem TUSAŞ envanterinde bulunan bir helikopterin uçuş testleri sırasında motorundan toplanan sıcaklık verileri üstünde eğitilip test edilmiştir.

Anahtar Kelimeler: Helikopter, Uçuş Test, Enstrümantasyon, Veri Güvenilirliği, Veri Tahmini

TEŞEKKÜR

Hayatım boyunca, şartlar ne olursa olsun desteklerini her zaman hissettiğim, bugünlere gelmemde, başardığım her şeyde en temel sebebim olan, haklarını asla ödeyemeyeceğim sevgili aileme;

Tez yazım sürecinde desteğini hiç esirgemeyen, her konuda tecrübelerini aktararak yardımını hiç kesmeyen danışmanım Dr. Öğr. Ü. Burkay GENÇ'e;

Tez için bana olanak sağlayan başta FTI ailesi olmak üzere tüm TUSAŞ Helikopter Başkanlığına;

Önceki çalışmalarım olduğu gibi tezimde de yardım eden ve ilerideki çalışmalarında da destek alacağım "Lifelong Project Buddy"im Serkan İSLAMOĞLU'na;

teşekkürü bir borç bilirim.

Barış Coşkun

Haziran 2021, Ankara

CONTENTS

LIST OF TABLES

LIST OF FIGURES

SECTION 1

INTRODUCTION

Helicopter is an aircraft that uses rotationary wings to provide lifting power to vehicle. Main rotor on top of the aircraft generates torque required for flight and tail rotor generates thrust against the torque generated by main rotor for controllability. By using these two rotationary wings, it is possible for helicopter to fly in all three direction (horizontal, vertical, lateral) and even hover which means hang in air with zero speed. Although helicopters are not capable of flying as fast as planes, due to their maneuver capability, they are widely used for both civilian and military purposes [1].

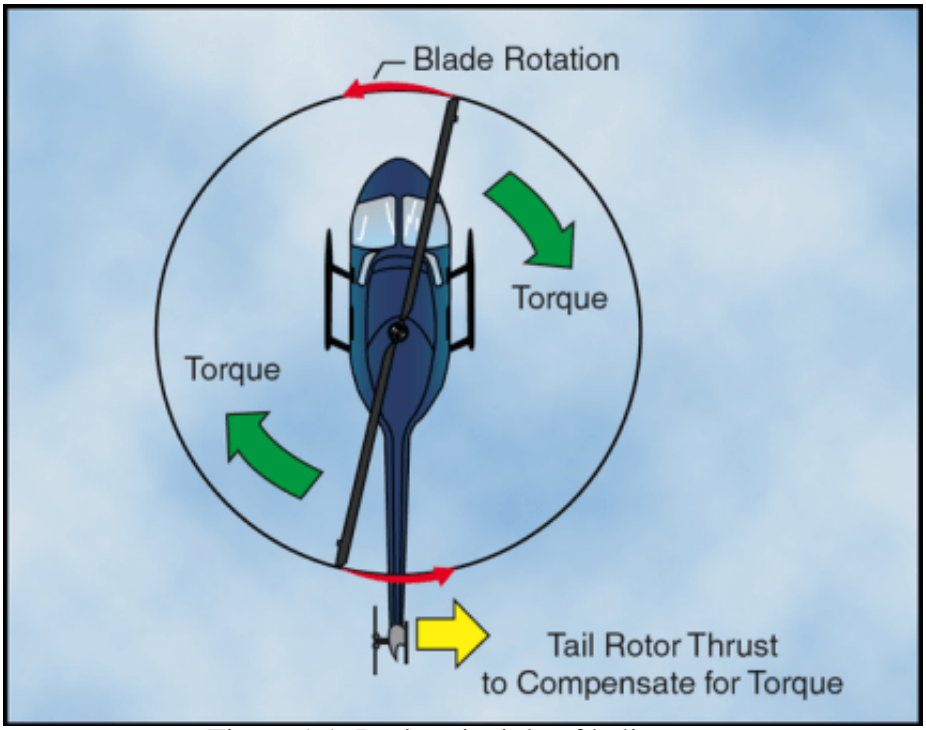

Figure 1.1: Basic principle of helicopter

As expected, designing a helicopter is a challenging task for engineers since there are numerous dynamic, rotational and electronic system that shall run in synchronously. All possible analysis and simulations are conducted in digital environment in order to make the design as perfect as possible. However, at the end of the day, actual flight test of a prototype is unavoidable to evaluate the design due to the production tolerances or unexpected systems interface problems [2].

Flight test campaign of an aircraft is a step-by-step process due to unpredictable risks caused by system or interface problems. So that, testing process starts from the most possible safe scenario while monitoring the helicopter by flight test instrumentation(FTI). Flight test instrumentation includes analog sensors that converts the physical magnitudes such as vibration, displacement, temperature etc to voltage which could be measured by data acquisition units and digitized with Analog to Digital Converters. Also FTI systems are capable of collecting digital messages such as Ethernet, RS232/422, Arinc429, Mil-Std-1553 which are generated by avionic devices on helicopter to maintain the communication between each equipment and also between pilot to aircraft [3]. In order to monitor the aircraft during any flight test, the data gathered by FTI system is transmitted to a ground station with telemetry by RF technology [4]. Field experts are monitoring the data transmitted from the aircraft in real time in the ground station in order to predict any kind of unexpected accident and interfere the test beforehand.

Figure 1.2: Helicopter crash during flight test

It is common to have maintenance or modifications frequently on prototype helicopters in flight test phase. During these activities, it is highly possible that any sensor gets damaged, dislocated or even miscalibrated without notice, because sensors are generally small compared to other parts of helicopter and unguarded since they are located at the surface. This kind of unexpected modification on sensor could lead to noisy or even completely meaningless data. The consequences of altered data could be catastrophic like reconducting the test which is expensive or even misinterpret the helicopter design.

Thanks to the latest developments in last decades, predictive models are capable of regenerating the missing and noisy data with high accuracy. However, for each system, the models shall be trained in a unique way to make them work as expected and provides a fresh challenge. In this thesis, we focused for temperature data from an aircraft produced by Turkish Aerospace. Two independent flight test data used in this thesis where helicopter configuration and test contents are equal. By training the model on one flight test and test them on another flight test data, we showed that it is possible to detect and correct different types of fault. We implement different predictive models such as Artificial Neural Network [5] and Decision Tree [6] for these prediction tasks and wide range of noise levels which indicates any kind of unexpected alternation on sensors could be detected.

In this thesis, we aimed to implement a predictive system that recovers the invalid data gathered by instrumentation on helicopter during flight test. For proof of concept, temperature measurements on helicopter engine is considered. In a scenario where a thermocouple is damaged, temperature readings from this sensor would be misleading and as a result, either the test would be cancelled which is waste of time and money or engine would be invisible for analysis which could make a possible accident invisible. In these kind of situations, field experts try to reproduce the data either using linear assumption or thermal model of engine using other measurements. These models are dependent to the complexity of implementation. Also there are some architectures in literature that uses predictive models in this purpose that are explained in Literature Search section in detail. The proposed models generally use one specific predictive model and test it on synthetic data. In the proposed architecture, the overall system is divided into two subsystems, one for predicting the amount of noise on selected

measurement and the other one for predicting actual value. For each model different kind of predictive models with optimal hyperparameters are designed and tested on actual flight test data with all possible noise cases.

SECTION 2

LITERATURE SEARCH

Predictive models are widely used in both academic and industrial projects. Predicting or classifying any data from the remaining information is valuable for both time and money saving. One of the most exciting studies was conducted by NASA in 1991 [7]. It is stated by inflight sensor group of NASA, dynamics of a propulsion system of a Space Shuttle is complicated is not well-known. In order to monitor the system, numerous instrumentation devices are applied which is generally more than it should be. Also, for some critical cases, more than one measurements is required to identify the sensor health by controlling the consistency between sensors.

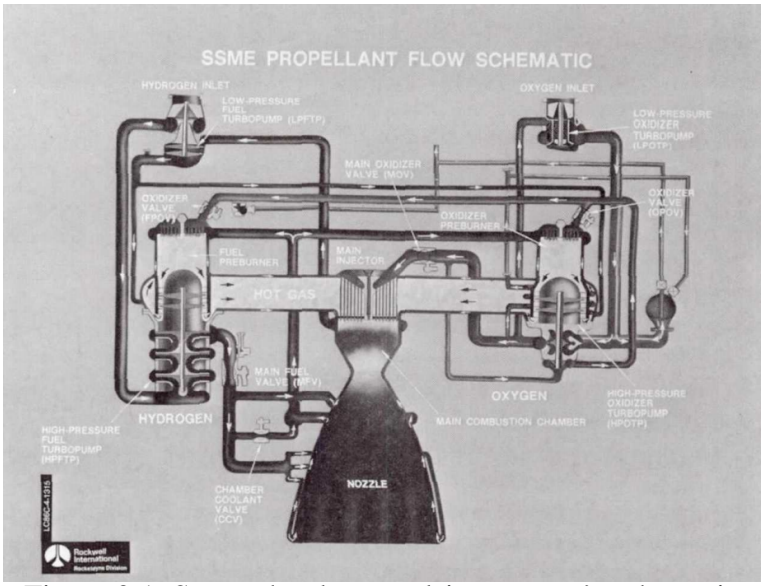

Figure 2.1: Space shuttle propulsion system's schematic

Even though there are numerous measurements as a precaution, an engine failure happened during simulation and virtually the space shuttle was destroyed. More so-

lemnly, there were numerous incidents during actual test of space shuttle where temperature readings were failed which caused component damage or unnecessary test cancellations. In order to improve the reliability of measurements, Kalman Filter was already used for detecting and isolating the sensor failures. By this research, a supervised predictive model, Neural Network is designed and tested that could replace the Kalman Filter for detecting and predicting feature.

As it could be seen from Figure 2.1, there are several subsystems in a propulsion system that shall be monitored. That requires high amount of measurements with different characteristics such as temperature, flow , pressure etc Figure 2.2. In order to decrease the complexity of the problem, the most critical subset of measurements are considered. Moreover by analysis and dynamic model of the propulsion system an "influence map" of measurements was constructed Figure 2.3.

P6:	Main Combustion Cooling Pressure
T6:	Main Combustion Cooling Temperature
Qfd1:	Low Pr. Fuel Pump outlet flow, in volume
Pfd1:	Low Pr. Fuel Pump outlet Pressure
Tfd1:	Low Pr. Fuel Pump outlet Temperature
Pfd2:	High Pr. Fuel Pump exit Pressure
Tft2d:	High Pr. Fuel Turbine Downstream Pressure
Sf1:	Low Pr. Fuel Turbopump Speed
Sf2:	High Pr. Fuel Turbopump Speed
Pc:	Main Combustion Chamber Pressure

Figure 2.2: Selected measurements

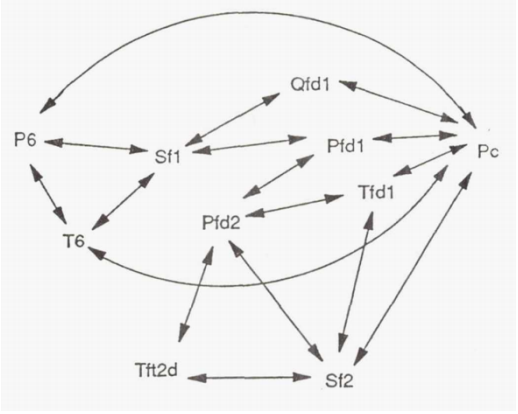

Figure 2.3: Influence map of selected sensors

For detecting the failure of a sensor, a feed forward fully connected neural network with two hidden layers each with 30 hidden nodes designed. The neural network generates a vector of ten outputs where each of them is the confidence metric for a measurement between 0 and 1, 0 being failure of the sensor and 1 is fully confident measurement.

In this study it was assumed that only one measurement would be failed and the second neural network is responsible for recovering the failed sensor's value from the remaining ones. Since the failed sensor could be detected by first neural network, second neural network takes remaining 9 measurements as an input set and generate

the output corresponding the the recovered measurement. For this purpose, neural network with two hidden layers with 30 hidden nodes per each trained.

For training the networks, start up of the propulsion system as chosen as training set which is around 4 seconds with 1 second steady state at the beginning. The 3 seconds of measurements with 50 Hz sample rate is enough for the networks to be trained since it is nearly covers all the operational points. During this training phase, one of the 10 selected data is selected randomly and Gaussian Noise with 0 mean and standard deviation of 1.5× "valid range of sensor" is applied. This "valid range of sensor" is defined by field experts and historical data of the sensor itself. By using the approach described, first neural networks is trained for detecting sensor failure with the Error vs Iteration graph as in Figure 2.4.

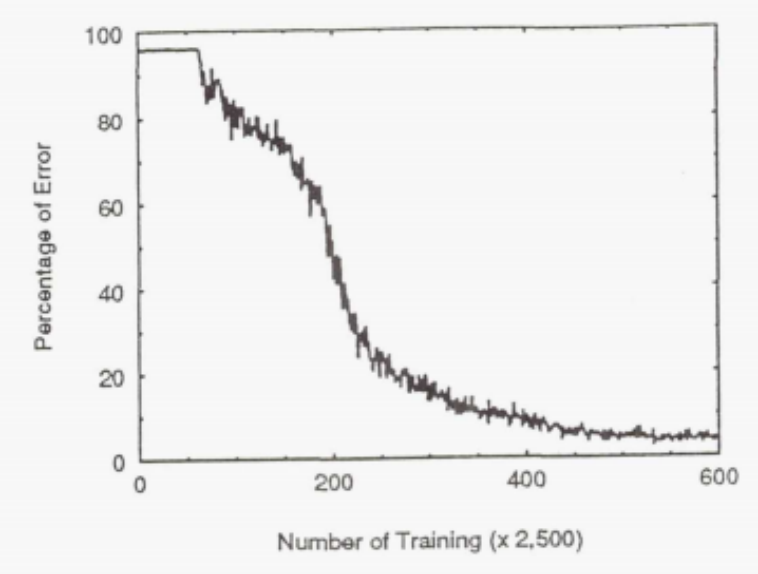

Figure 2.4: Training error of first neural network

At the same time, one of the selected 10 sensor measurements is selected and second neural network is trained in a way to predict the selected measurement from remaining 9 of them. This neural networks are tested in digital simulation environment for two cases where in case-1, HPFTP speed sensor is failed at $T = 7$ and case-2 MCC Pressure Sensor failed at $T = 8$. The results are demonstrated in Figure 2.5 and Figure 2.6.

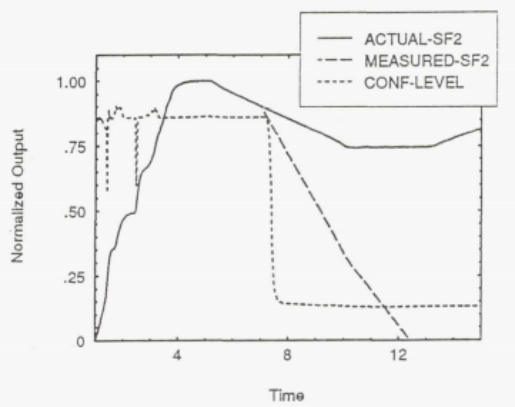

Figure 2.5: Test case-1 results

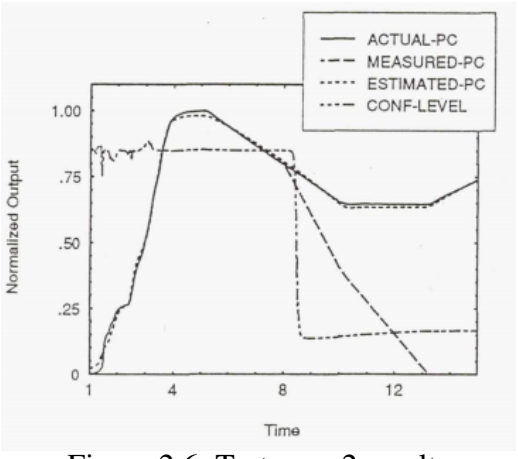

Figure 2.6: Test case-2 results

Another experiment about temperature reading validation and correction on a Rocket engine was conducted by Ezhilrajan et al at 2020 [8]. Cryogenic Propulsion System of Rocket Engine has its own controller for adjusting the mixture ratio and thrust which are vital for engine operation. Two temperature readings are acquired from Main Pump Delivery with a certain protruding length that are used together for certainty and used as feedback for control system. In case of invalid temperature readings at the sensors, controller may supply less propellant than required which decreases the payload and even in some cases, unpredictable mixture ratio may lead to catastrophic accidents.

There are mainly 3 types of sensor data validation and correction methods. First model is online parameter identification techniques which are called model based fault diagnosis. Second model is rule based which depends on heuristic algorithms and expert knowledge. Last model is Neural Networks that are trained with proper set of data. Ezhilrajan et al proposed a more intelligent hybrid algorithm which uses fuzzy logic and neural networks together called Adaptive Neuro Fuzzy Inference System (ANFIS). ANFIS is basically a feed forward neural network with supervised learning algorithms. In order to train the network, one of the most popular training algorithm of neural networks, Gradient Descent is used. Since, gradient descent algorithm has a tendency to get stuck in local minima, it is combined with one of the hybrid learning algorithm, least square type method. The architecture of ANFIS is visualized in Figure 2.7.

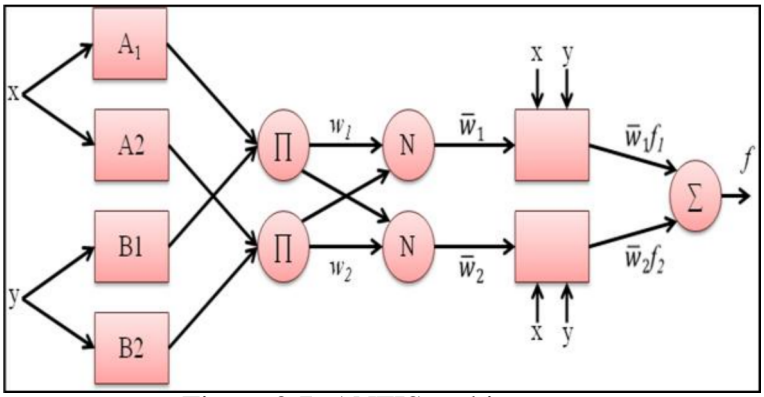

Figure 2.7: ANFIS architecture

There are 5 layers in ANFIS architecture and each of them has individual specific task. First layer is responsible for mapping the input to relative membership function. Second layer is responsible for firing strength using T-norm operation in order not to be trapped in local minima. Third layer normalizes the effect of firing from second layer. Fourth layer applies the determined rules and in the last layer the output is computed.

This hybrid approach is applied for the case where temperature reading is inconsistent for first 60 seconds. To create a Temperature Sensor Data Validation (TSDV) model, the procedure defined by block diagram in Figure 2.8

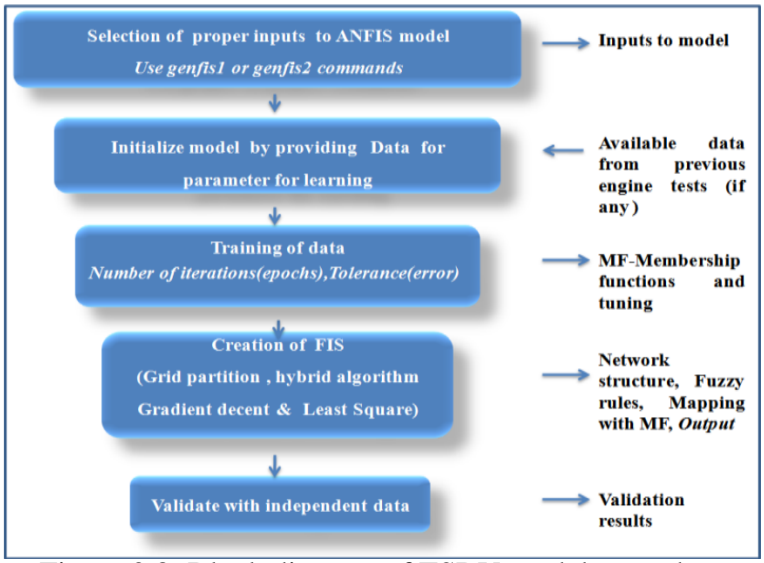

Figure 2.8: Block diagram of TSDV model procedure

In order to select proper inputs for ANFIS TSDV model, influence map is generated

as in Figure 2.9. From the influence map and dynamic analysis of models, 4 linearly dependent parameters to LOX Main Pump Delivery Temperature (TOPD) which is the ouput of the model have selected. These 4 parameters are; Main Turbo Pump Outlet Temperature (MTPOT), LOX Booster Pump Outlet Temperature (LBPOT), LOX Main Turbo Pump Delivery Pressure (LMPDP) and Chamber Pressure (PC).

To train the ANFIS, 800 data pairs are provided to model to develop a Fuzzy Interference System (FIS) and ANFIS generated 900 parameter where 876 of them are linear and remaining 24 of them are nonlinear. With the 193 nodes in neural network state and 81 linguistic rules generated by back propagation algorithm, overall model looked like in Figure 2.10.

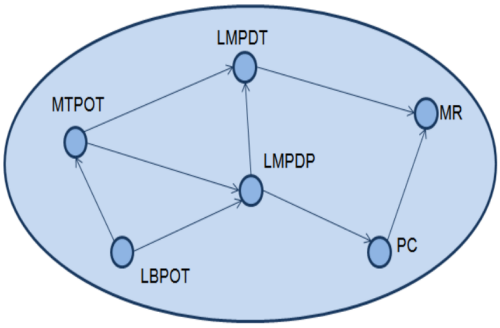

Figure 2.9: Influence map

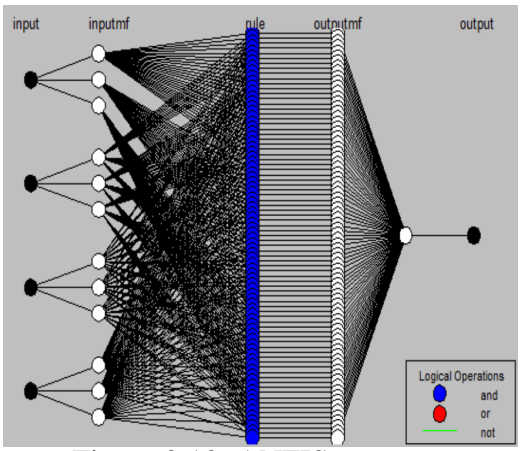

Figure 2.10: ANFIS structure

In the testing scenario, temperature reading is 4.5 K higher than the actual value and it takes 60 to come to acceptable error range. This indicates the wrong assembly of the sensor with inappropriate protruding length. The actual measured, expected and predicted temperature and Mixture Ratio by model is illustrated in Figure 2.11 and Figure 2.12.

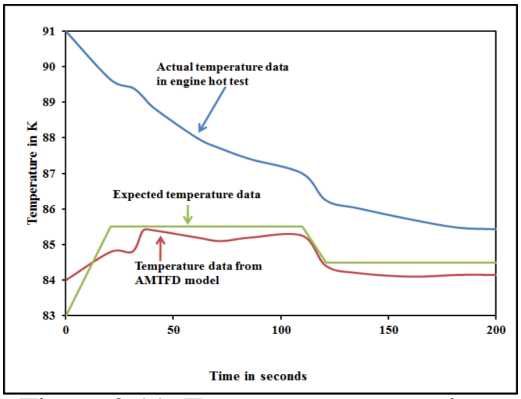

Figure 2.11: Temperature comparison

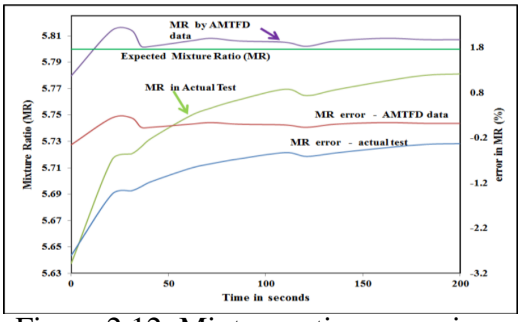

Figure 2.12: Mixture ratio comparison

Ogaji et al increases the complexity of fault detection of an engine using Neural Networks [9]. It is stated that engine components are more reliable than instrumentation although any fault on an engine component requires more time to recover. In order to predict fault beforehand, a system consist of different neural networks is designed. The aim of the system is detecting single or dual sensor failure and single, dual or multiple component fails. The architecture is designed as in Figure 2.13.

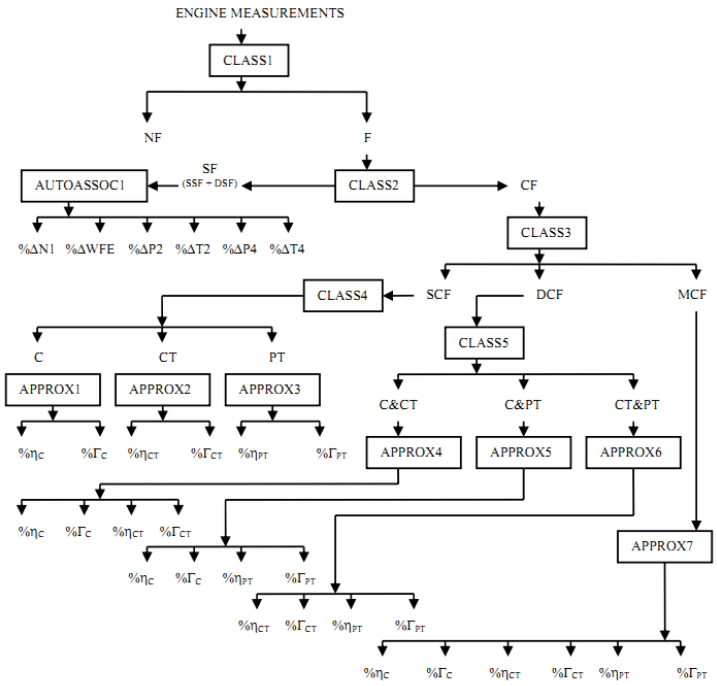

Figure 2.13: Diagnose architecture

In the diagnose path in Figure 2.13, CLASS1 corresponds to the first classification of fault or not fault. If a fault detected on the system CLASS2 detects the cause of

fault like sensor or component fault. AUTOASSOC1 responsible for determining the amount of noise or bias in sensors. CLASS3 networks detects the number of faulty components if the reason is component fault. CLASS4 detect the faulty component among 3 possible options; Compressor, Compressor Turbine and Power Turbine.

For each CLASS defined in diagnose path, an individual probabilistic neural network trained. PNN networks could train itself in less than 2 minutes which could be considered as "no training required". The accuracy of each CLASS is defined in Tab 2.1

Network	**Results (%)**		
CLASS1	NF 100	F 99.9	
CLASS2	SF 100	CF 99.7	
CLASS3	SCF 99.1	DCF 90.1	MCF 76.3
CLASS4	C 100	CT 100	PT 100
CLASS5	C & CT 98.6	C & PT 96.8	CT & PT 97.2

Table 2.1: Accuracy of networks for each CLASS

All classification accuracy values are satisfying except CLASS3 where DCFs are misclassified as SCF or MCF. This is because the affect of these faults on instrumentation data is similar to each other. In order to overcome this problem, Neural Network with $6 - 35 - 35 - 3$ architecture is trained with resilient backpropogation and tan sigmoid transfer function on all nodes. The accuracy for CLASS3 increases to 98.85 for SCF, 95 for DCF and 90.44 for MCF.

Instead of a path of diagnosis method, authors proposes a single neural network called CLASSΩ as in Figure 2.14. The classification accuracy values are; 100 for NF, 99.62 for SF, 99.82 for C, 98.42 for CT, 98.42 for PT 89.06 for C & CT, 87.76 for C & PT, 86.72 for CT & PT, 76.02 for MCF. These classification results are similar to previous architecture (Tab 2.1). Optimal neural network architectures are used for each APPROX and they are defined in Tab 2.2

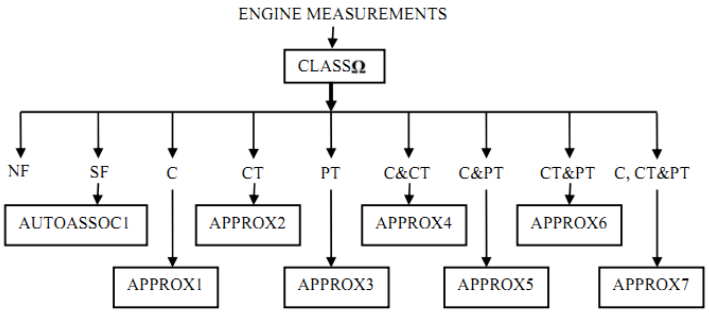

Figure 2.14: Single CLASS architecture

Network	Design	Size	MSE (Training)	MSE (Test)
APPROX1	RB	6-15-15-2	0.009	0.01
APPROX2	RB	6-10-10-2	0.003	0.003
APPROX3	RB	6-10-10-2	0.002	0.003
APPROX4	RB	6-30-30-4	0.032	0.032
APPROX5	RB	6-35-35-4	0.018	0.018
APPROX6	RB	6-30-30-4	0.018	0.018
APPROX7	RB	6-40-40-6	0.137	0.146

Table 2.2: Details of APPROX neural networks

The authors summarized the paper by explaining the pros and cons for each method. Gas Path Diagnosis Path is open to improvement by adding expert decision or fuzzy logic between steps. On the other hand single and complex neural network could use other data sources such as vibration or oil consumption which could increase the accuracy.

Sensor failure detection and data recovery is essential for almost all part of aircraft. Napolitano et al designed a Sensor Failure Detection, Identification and Accommodation (SFDIA) system for flight control systems [10]. The overall architecture of SFDIA is give in Figure 2.15. Flight Control System is a closed loop system consist of actuators, components and sensors where the vulnerable part of whole system is sensors. Any failure in the sensor would lead to failure of control system so that researches on SFDIA become more and more important. In the paper, failures are categorized into 2 types, hard and soft failures. Hard failures are basically complete destruction of sensors which are easy to detect and directly downs the whole flight control system. Soft failures are drift or bias in data from sensors which could not be detected easily and would lead to catastrophic accidents. Napolitano et al proposed

two models for SFDIA, Kalman Filter and Neural Network based.

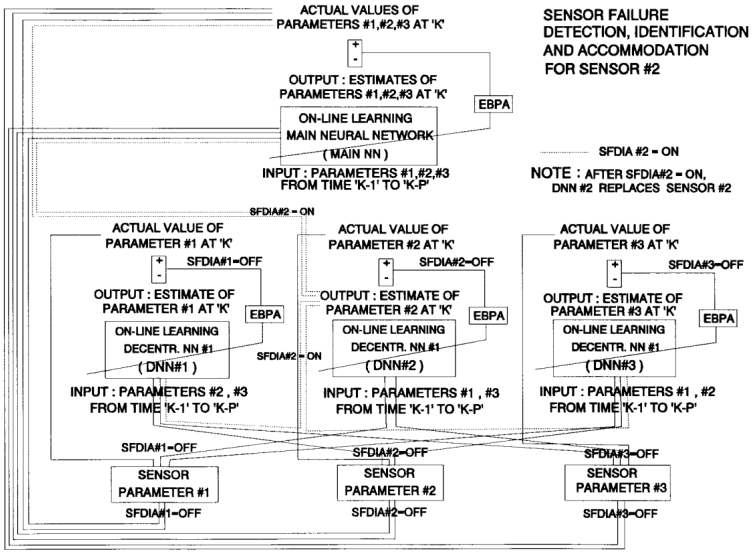

Figure 2.15: SFDIA architecture

It is proven that Neural Networks have numerous advantages for regenerating the data. It is possible that Neural Networks could be trained online to estimate the data afterwards. Also it is possible for neural network to learn nonlinear relation of data no matter number of input and outputs even for a single hidden layer network. Due to its modularity and parallel architecture, neural networks could be embedded into a hardware easily with fault tolerant and high speed implementation.

Unfortunately, designing a neural network is a challenging task especially for the cases where network should train itself and predict values in real time. Tradition Neural Network design with Hidden Nodes with sigmoid activation functions and Gradient Decent back propagation algorithms could not compete in real time. So that authors make the each hidden node capable of rearrange the slope of sigmoid function which increases the training rate for specific hidden node. This algorithm called extended back propagation and make the real time training possible. Also during the training phase, authors defined a threshold for error rate between actual sensor reading and neural network prediction. If the error rate exceeds the threshold, neural network stops training itself and replaces the sensor reading with the prediction. If the real time training is burden to system, the sample rate would be decreased for training and previous test data are used for pretraining the neural network.

Kalman Filter is one of the most popular and robust mechanism for predicting upcoming data using state transition matrices. Since Kalman Filter are relying on linear matrices, for nonlinear relations, extended Kalman Filters are required where the errors are linearized. However, these extensions on Kalman Filters decreases the processing speed and lead to unbiasity so that only linear dynamic conditions are considered for Neural Network and Kalman Filter comparison.

For testing these two models, high altitude unmanned aircraft used by NASA for Mission to Planet Earth program is used. For a sensor of pitch rate gyro, aircraft is simulated and two models are trained in real time. In the simulation, at $t = 29$, sensor failed and the response of models are demonstrated at Figure 2.16.

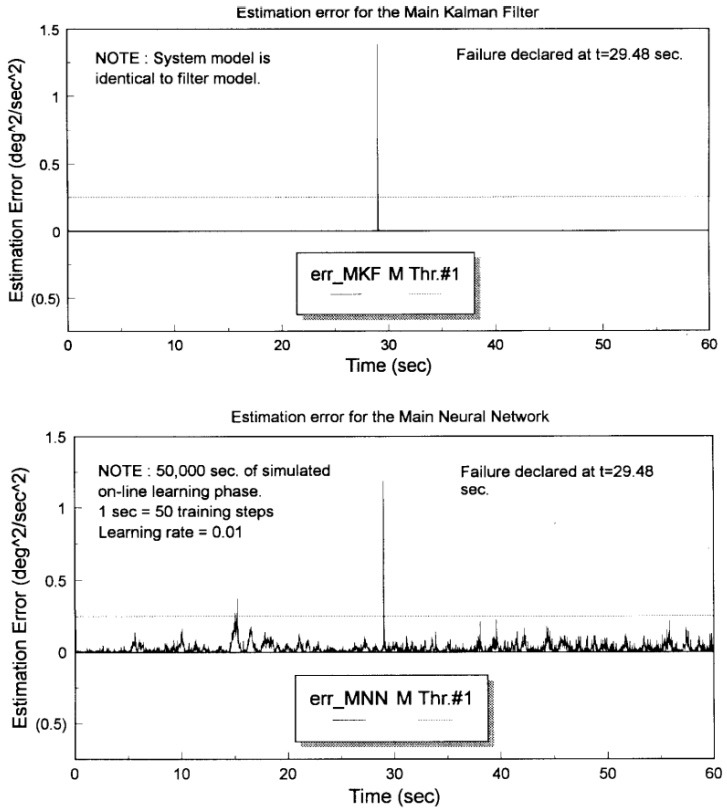

Figure 2.16: Kalman Filter vs Neural Network

Authors tried 6 different failure types; small/large instantaneous bias, small/large bias with small/large ramp with 5 configurations according to the linearity level caused by measurement noises and discrepancies between simulation and actual aircraft. The detailed results are provided in paper and as a results authors stated that Kalman Filter provides more robust results than Neural Network. However, for the cases when

linearity in system decreases by the noises and discrepancies, Neural Networks become superior. Also for linear noises such as ramp failures, Kalman Filters could not detect while Neural Network could realize the failure with not a perfect accuracy.

In literature, predictive models are not only used for fault in system. Huang et al used Neural Network for vibration diagnosis of a motor [11]. Airplane engine's physical models are generally high order (25000^{th}) finite element models. Any undesired mass on turbine motors generates unbalance during the rotation which could be located and identified by using vibration analysis and engine model. However, in order to describe the model clearly, authors defines a 2^{nd} order vibration model with rotational disk as in Figure 2.17.

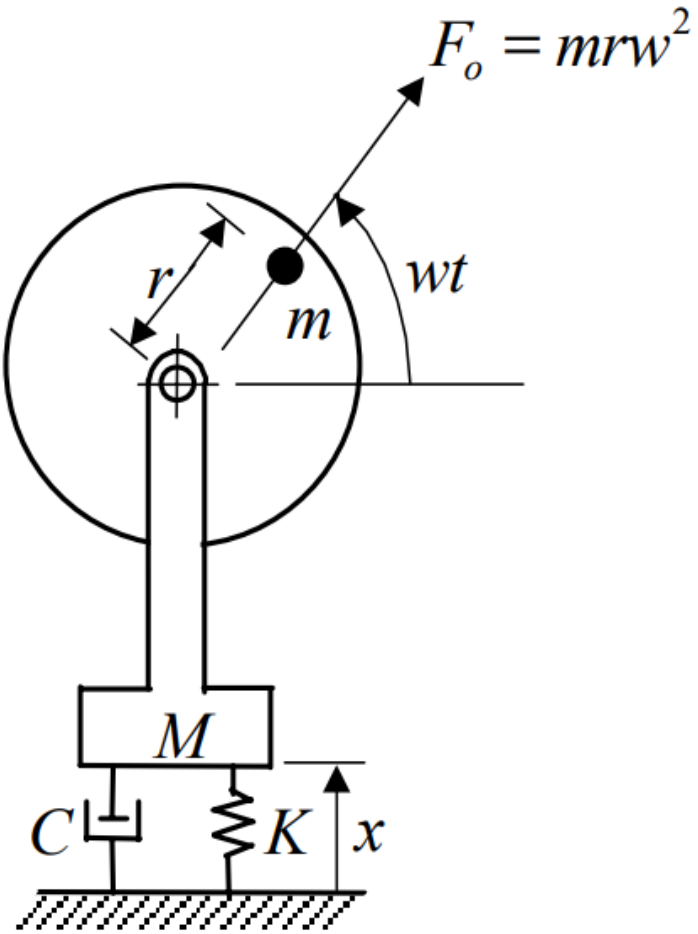

Figure 2.17: Simple model for vibration analysis

A neural network with a single hidden layer is trained by data calculated by physical formula of simple model. The mass location remained fixed during training and neural

network is trained to determine the amount of mass using frequency response from the measurements. The test error is within 0.001 inch-ounce which encourage to more complex experiment.

For more realistic experiment, authors used MSC/NASTRAN model of an turbine motor with approximately 26000 degrees of freedom. As a experiment, one ounce-inch unbalance mass located on 3 different locations; fan, turbine stage 1 and turbine stage 5. Frequency responses are calculated from 3 different measurement points; bearing 1 near to fan, inner shaft bearing and bearing 6 near turbine. In order to predict the amount of unbalance mass, 3 different learning methods are used; back propagation, Extended Kalman Filter(EKM) and Support Vector Machines(SVM). Each model is trained with data from digital model in order to predict the source and amount of unbalance map. The training dataset consist of 250 frequency bins from $0.2Hz-50Hz$ for each location with 6 different mass value $0.0, 0.2, 0.6, 1.0, 1.4 and 1.8 inch-ounce$. The networks structures and error for mass values are represented in Tab 2.3. SVM and EKF models shows promising results although required time for training or testing for these models are higher than BP.

Network Architecture	**Learning Method**	**Training Error Range**	**Testing Error Range**
1500-30-3	BP	0.07	0.07
1500-30-3	EKF	0.05	0.05
1500-30-3	SVM	0.03	0.05

Table 2.3: Results for 3 different model

Neural networks are used for increasing the sensor confidentiality for almost all kind of experiments. Yang et al used Radial Basis Function (RBF) Neural Network to increase the accuracy of motion control by using regressing several measurements [12]. Instead of using the measurement data blindly, they implement a RBF neural network that calculates the regression weight of sensors from their attributes as it is sketch in Figure 2.18. By using the fused data for motion controller instead of measurements, they eliminated the possible errors that could be caused by measurements or sensor itself.

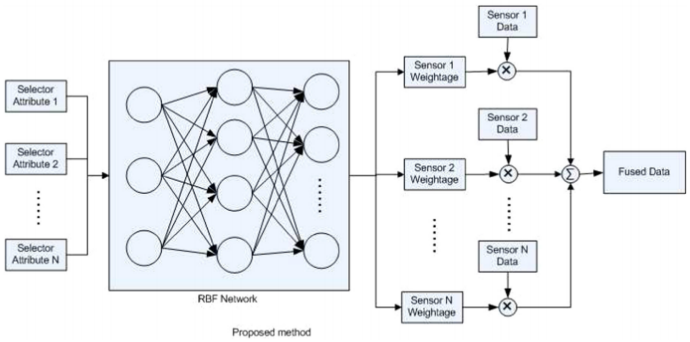

Figure 2.18: Proposed network

Before the weighting operation, all measurements are calibrated according to a measurement from a laser interferometer with nanometer resolution. In this paper, it is proposed that instead of linear calibration, a RBF neural network could be used for weighting the measurements by considering the measurement and the noise level as selectors. Since RBF neural networks are able to map the inputs to a output, it could be easily trained in a way to calibrate the sensors according to the pivot measurements from laser interferometer.

As a test setup, an analog velocity sensor and magnetic encoder are attached to a DC motor. As it is known that analog sensors have infinite resolution although their sensitivity is limited. So that, for lower velocities, it is expected that analog sensor have higher weight while for higher speeds digital magnetic encoder is going to be more accurate. The calculated weights for different noise levels and different velocities are provided where k_1 being the weight for analog velocity sensor and k_2 is for digital magnetic encoder.

k_1 Selection for different velocities and noise levels.

Noise (%)	Velocities (m/s)						
	0.04	0.10	0.20	0.30	0.40	0.50	0.60
5	0.87	0.875	0.61	0.46	0.25	0.29	0.00
10	0.87	0.885	0.62	0.47	0.255	0.29	0.00
50	0.89	0.98	0.585	0.405	0.30	0.265	0.00
100	0.915	1.00	0.665	0.48	0.185	0.31	0.00
300	0.825	1.00	0.48	0.285	0.025	0.105	0.00
500	0.375	0.38	0.205	0.01	0.005	0.055	0.00

Figure 2.19: Weights for analog velocity measurements

k_2 Selection for different velocities and noise levels.

Noise (%)	Velocities (m/s)						
	0.04	0.10	0.20	0.30	0.40	0.50	0.60
5	0.135	0.13	0.39	0.54	0.75	0.705	0.995
10	0.135	0.12	0.38	0.53	0.745	0.705	0.995
50	0.115	0.025	0.42	0.6	0.7	0.73	0.995
100	0.09	0.01	0.34	0.525	0.82	0.69	0.995
300	0.185	0.035	0.54	0.73	0.98	0.895	0.995
500	0.65	0.665	0.82	1.00	1.00	0.945	0.995

Figure 2.20: Weights for digital magnetic encoder

More testing scenarios are explained in detail by changing other possible parameters in test setup. To sum up it is stated that using a RBF Neural Network to fuse different measurements into single generates more accurate result than any other sensor could

achieve independently.

Mese et al also worked on a controller for a motor using neural network [13]. Different from [12], they designed a Artificial Neural Network that could predict the position of a motor from not linearly correlated measurements like flux linkage and current. By using sufficient enough data, an ANN could predict the position of motor from flux and current without the actual position sensor. Having a ANN instead of position sensors would be more advantageous in terms of money and accuracy.

By using the actual motor with position sensor, enough data collected for training the ANN. The optimal hyperparameters are found by trial-and-error method and one hidden layer with 10 hidden neuron found to be good balance between accuracy and complexity. The training results are demonstrated in Figure 2.21.

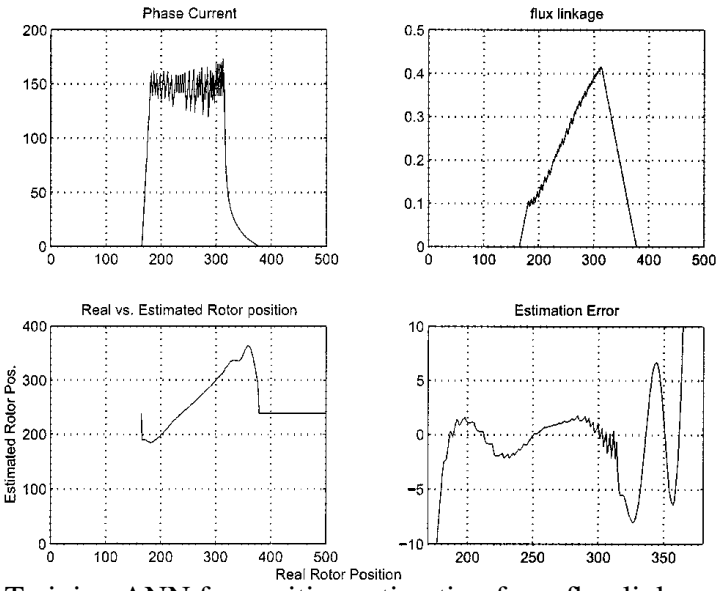

Figure 2.21: Training ANN for position estimation from flux linkage and current

However, for electrical purposes, flux measurements is not accurate enough. So that, authors designed another ANN with one hidden layer includes five hidden neurons for flux prediction from given current and position. The difference between actual and predicted flux is occurred due to the error in position estimation. This error could be used for correcting the position estimation as a feedback in control loop. The decrease in prediction errors are shown in Figure 2.22.

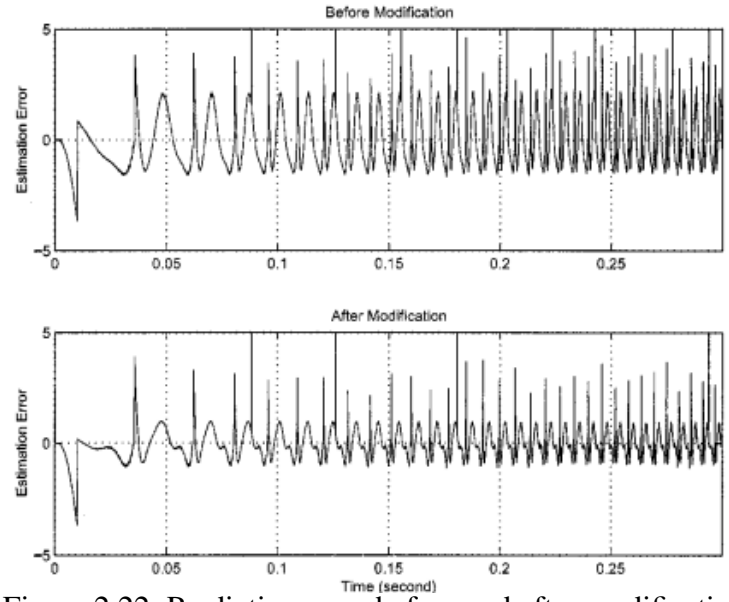

Figure 2.22: Prediction error before and after modification

This proposed method have tested in both real-time prediction by designing a circuit that includes the ANN parameters as lookup table and in digital environment using simulation tools such as MatLab. The experiments showed that the error is bounded on $[-5°, +5°]$ but mostly around $[-2°, +2°]$.

It is also possible to detect sensor failure using other algorithms in literature. Alippi et al propose Hidden Markov Model for sensor failure detection with Cognitive Layer for classification for sensor networks [14]. Authors stated that HMM is able to detect the anomaly in a distributed sensor network. HMM mines the relation between any given set of values and generates a relation function between them. If any data not acceptable by learned function generates an alarm in system. These alarms are occurred due to 3 main reasons; model bias, change in environment and sensor failure. The models in literature designed with HMM are assumed model bias is negligible and not able to differentiate whether the alarm generated by change in environment or sensor failure. In this research, authors propose and additional Cognitive Layer above HMM for error classification as in Figure 2.23.

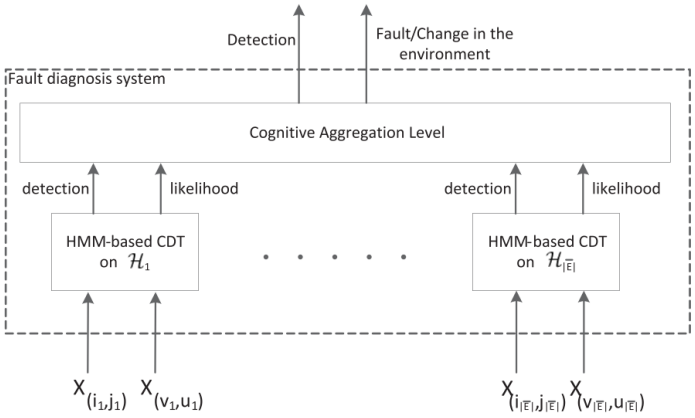

Figure 2.23: Proposed architecture

Cognitive Layer is responsible for classification of sensor failure and environment change. Basically, if any of HMM generates an error detection, Cognitive Layer controls other HMM models related to the detection and decides according to the likelihood of error by considering all HMM model on network. This proposed architecture is tested in various real sensor networks. The detailed results for all experiments are provided in paper.

Sensor fault detection algorithms are widely used in many diverse areas. Liao et al studied sensor fault detection and recovery (FDR) for automated driving system [15].

For FDR, authors designed a Multitask One-dimensional Convolutional Neural Network (MONN) as in Figure 2.24. Firstly encoder architecture used for information extraction from input data. FR-Subnet used for data recovery and decoder used for fault detection. These subsystems are unique Convolutional Neural Networks that details and optimizations are described in detail in paper. From digital simulation program called CarSim used for gathering training data and used for training MONN.

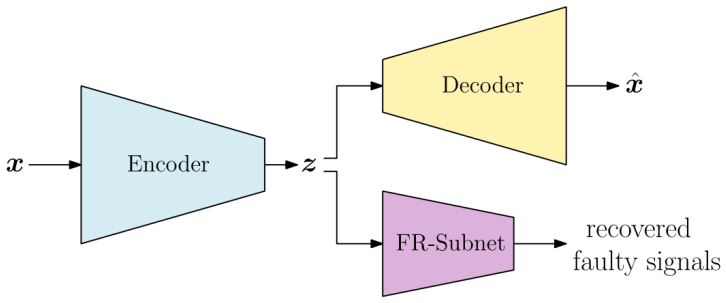

Figure 2.24: MONN architecture

Authors compared their MONN architecture with other state-of-art designs in literature. The detailed Fault Detection accuracy values are described in Tab 2.4 and speed of processing results are in Figure 2.25. MONN architecture clearly bests other base models. In addition to these results, authors tries their model with different implementations and also under different testing conditions.

Model	F_1 **Score**	F_2 **Score**	$F_{0.5}$ **Score**	**AUC**
IF	0.71	0.753	0.672	0.704
IF (PCA)	0.796	0.795	0.797	0.85
OCSVM	0.817	0.757	0.874	0.851
OCSVM (PCA)	0.747	0.803	0.697	0.819
LOF	0.88	0.851	0.915	0.92
AE	0.882	0.859	0.881	0.905
ICS	0.842	0.855	0.885	0.89
DSVDD	0.86	0.833	0.897	0.896
ALOCC	0.764	0.681	0.87	0.844
MONN	**0.906**	**0.885**	**0.928**	**0.93**

Table 2.4: Comparsion of FDR models

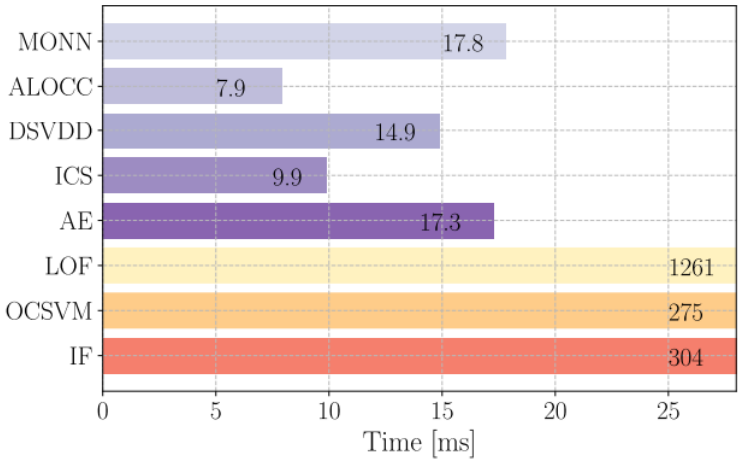

Figure 2.25: Time consumption for 2000 observations during test

SECTION 3

DATASET

Supervised predicted models are capable of recognising and predicting data according to their experience during training phase so that, it is essential to have confirmed data to train the models. Training algorithms are designed in a way to extract the information on the training dataset and adjust the model to recognise or predict any missing data from the information left. For this purpose, two identical flight test data from a helicopter produced by Turkish Aerospace is utilized where one of them is used for training while the other one is used for testing the trained model to measure its performance.

It is not feasible to train models that is able to predict any data gathered by FTI since there are thousands of parameter in an FTI system with different characteristics. Also it is predictable that not all the data in the system are informative about each other since there are different subsystems on helicopter. Due to these reasons, a subsystem of helicopter, an engine was selected and all the temperature data collected from the engine is used as dataset.

3.1 Thermocouples

As it is explained in introduction section, analog sensors are used for converting any kind of physical magnitude to voltage values which could be measured and converted into digital formats using FTI data acquisitions units with Analog-to-Digital Converters. For temperature readings, temperature-sensitive cable pairs are used which is called thermocouples [16]. According to the temperature sensitive cable type used within thermocouples, there are different types such as chromel-alumel pair used in

K type thermocouples.

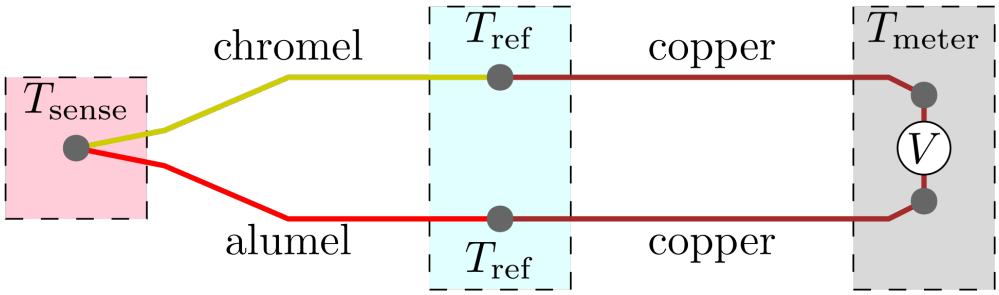

Figure 3.1: Thermocouple working principle.

Due to the Seebeck principle, each type of metal generates different voltage according to the level of temperature it is exposed as in Tab 3.1. Since the voltage generated by temperature difference could be calculated mathematically, it is possible to tabulate the measured voltage corresponding to temperature [17]. This tabulation is embedded into FTI data acquisition units and the measured voltage is converted into temperature value in real time.

Material	**mV at 100 °C**	**mV at 200 °C**
Alumel	-1.29	-2.17
Chromel	2.81	5.96
Constantan	-3.51	-7.45
Copper	0.76	1.83
Iron	1.89	1.44

Table 3.1: Seebeck effect on different type of materials

3.2 Instrumentation

Engines of helicopters are instrumented in order to keep track of vibration and temperature levels during flight tests. In this thesis, we focus on temperature readings from 11 independent thermocouples on an engine. All these thermocouples are located on critical parts of the engine. However, the locations could not be revealed and names of the sensors are replaced with generic names for confidentiality purposes. Sensors will be referred to as T_X where X is just the index of the sensor in the dataset.

Dataset	**Test Length**	**Number of Measurements**	**Number of Observations**
Training Dataset	2 hours	11	105342
Test Dataset	2 hours	11	113826

Table 3.2: Dataset Statistics

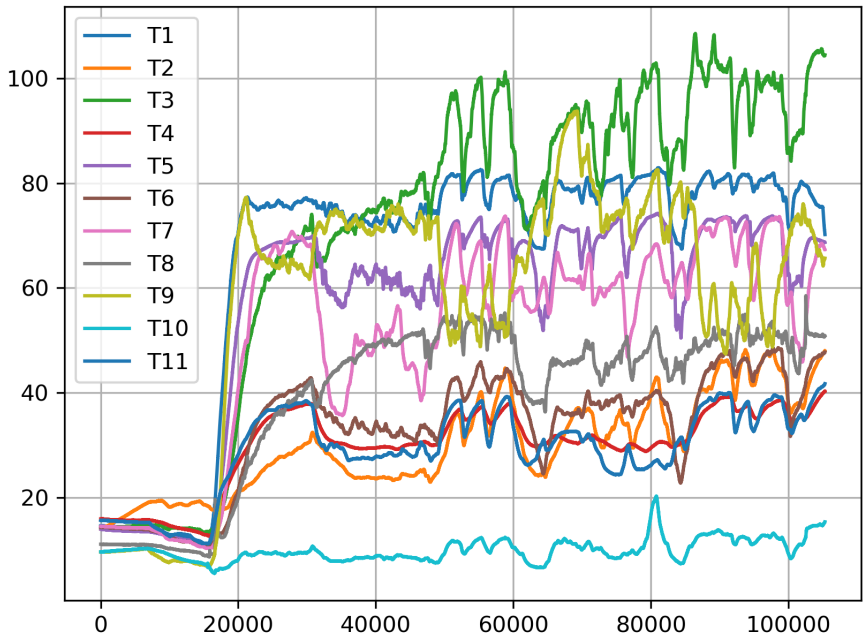

Figure 3.2: Temperature readings from 11 thermocouples at 16Hz for a period of 2 hours

Temperature values acquired from 11 independent thermocouples are visualized in Figure 3.2. Y axis corresponds to temperature values in °C measured by each thermocouple while X axis corresponds to timestamp. It is possible for FTI system to sample sensors at different rates which means how many observations are going to be collected within a second. Sample rates are decided according to the analysis requirement where if any high speed physical behaviour are occurred on the place sensor located, the sample rate shall be at least two times higher then aimed movement according to the Nyquist Criteria [18]. Due to the characteristic of temperature, 16 Hz is chosen as sample rate for thermocouples and as it can be seen in Figure 3.2, there are 120K observations for 2 hours flight test.

By looking Figure 3.2 only, it is possible to understand the general behavior of thermocouples. In the beginning of test, it could be seen that all T_X values are around 15 °C which is the outside air temperature during flight test. Around the 18000^{th}. time stamp, all T_X readings increase tremendously. This is the point where ignition started at that engine. However, T_{10} have completely different pattern. This is because the

location of thermocouple related to T_{10} is away from combustion chamber of engine. After the transient ignition state, it could be stated that all the T_X values are independent from each other since thermal effects such as cooling or heating is not identical on each part of engine.

Instead of repeating the similar observations for each sensor one-by-one, we chose T_9 as a pivot data for visualization and detailed explanation. Since the sensors are not trivially related to each other, Figure 3.3 shows how the sensors arbitrary behave with respect to T_9.

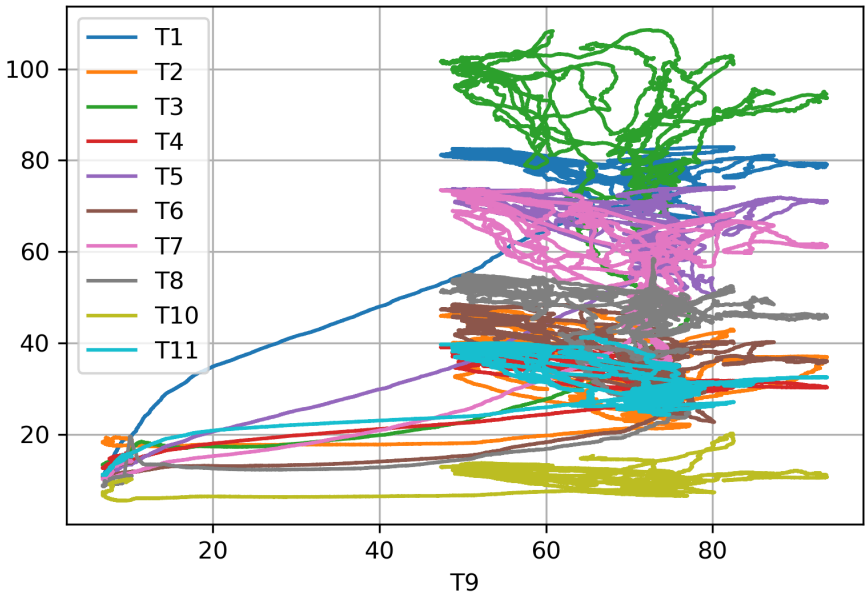

Figure 3.3: T_9 vs. other sensors

As expected, readings are related between from 20 °C to 50 °C which corresponds to beginning of the test to end of ignition phase. After the ignition, when the engine starts working stably, there is no correlation between temperature readings due to thermal model of engine itself.

3.3 Noise Generation

Supervised predictive models are trained according to previous experiences which is training dataset. However, in order to train models to be responsive for misreadings,

it is essential to have invalid readings with label in the dataset. Since the dataset we are using is from an actual flight test of a helicopter in Turkish Aerospace, it is known that data is valid. In order to generate invalid temperature readings, we add noise to sensors synthetically and label them according to the noise ratio.

There are many source that could alter the analog sensor reading from actual physical magnitude which could be divided into two groups according to their relative effect on data itself. First and most common one are noises generated by other sources such as EMI/EMC, Electrical Gaussian Noise etc. These kind of noises have completely random characteristic due to their nature. Fortunately, their signal level are either so low or they are instant spikes which could be easily detected or neglected [19]. In the second group, there are misreadings not caused by outside effects but caused by hardware malfunctioning on sensor or data acquisition units. These kind of misreadings alternates the data significantly and could lead to catastrophic results. This kind of misreading could make the evaluation during real time monitoring invalid which would stop the flight test needlessly which is expensive in terms of money and time. More seriously, altered data could lead to misinterpretation by analysis experts and makes the upcoming potential accident invisible.

In this thesis, we aimed to design a sequential system that could understand how much data is corrupted and recover it by using remaining data. This system would generate a confidence metric for data itself to prevent misinterpretation and make the test data usable for analysis even in sensor failure in reality.

We mainly focus on single point of failure during flight test where only one temperature reading is invalid. In order to create supervised predicted models for each 11 temperature reading, we modify the raw dataset 11 times, such that each single sensor reading is synthetically corrupted according to the following equation:

$$T'_X = (1 + \nu)T_X, \quad \nu \in [-1, +1] \tag{3.1}$$

where, ν represents the applied rate of noise with respect to the actual reading. By using ν value between -1 to +1, nearly all possible misreading scenarios are covered where from $T'_X = 0$ which is equivalent to complete destruction of sensor or data

acquisition unit itself to $T'_X = 2 \times T_X$ which is corresponds to serious miscalibration or dislocation of sensor.

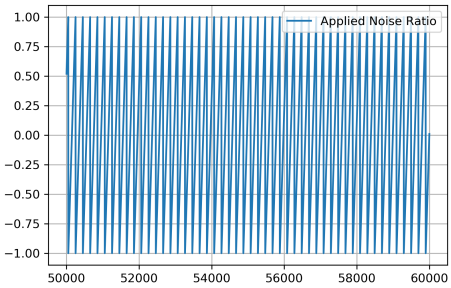

Figure 3.4: Applied noise ratio pattern.

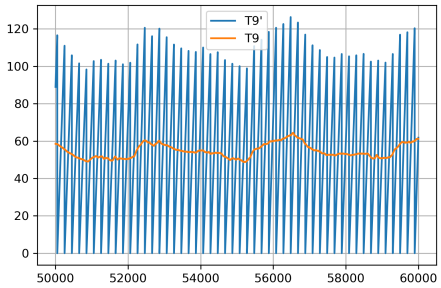

Figure 3.5: T_9 vs T'_9

In order to have a homogeneous distribution of noise over data, ν has been systematically shifted in the range from -1 to +1 with a step size of 0.01 as demonstrated in Figure 3.4 and the generated data is shown in Figure 3.5 for T_9. Both figures show the data in a limited time window for better presentation purposes. By using this noise generation principle, models could be trained in a way to recognise all level of misreadings without any bias.

SECTION 4

METHODOLOGY

The main aim of this thesis composed of two sequential task. Firstly, the amount of noise for a single T_X from given set of data is going to be predicted. After that, actual value of the T_X is going to be predicted using set of data and noise ratio predicted by previous system. Since there are eleven independent T_X in dataset and the main assumption is there would be single point of failure where only one thermocouple produce faulty reading at once, there are eleven individual predictive model responsible for each T_X for each task and this array of predictive models are called 'system'.

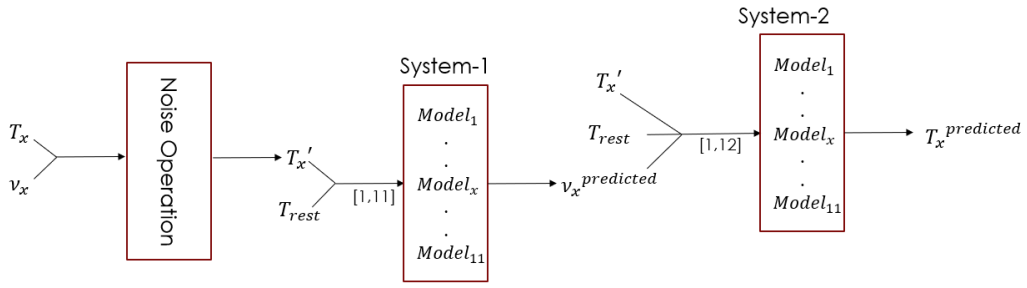

Figure 4.1: Overall Architecture

Whole proposed architecture is described in Figure 4.1. For each observation, noise value according to ν is applied on T_X that is symbolized as T'_X. System-1 is responsible for predicting ν from observations for each 11 independent measurements including T'_X instead of T_X. System-2 is trained for predicting T_X using observations of measurements and predicted noise from System-1.

4.1 System-1 - Noise Ratio Prediction

The first aim of this thesis is predicting the amount of noise of a single T_X in given set of data assuming that remaining T_X values are valid. In order to train predictive models to be capable of extracting noise ratio of a T_X from given set of data, training dataset shall include noisy data with proper labeling. As explained in Section 3.3, flight test data which is valid for sure, is synthetically modified for each T_X and used for training the predictive model responsible for corresponding T_X.

4.1.1 Model-1 - Artificial Neural Network

Artificial Neural Networks are one of the most popular supervised predictive model in recent years. Due to their high capacity of information extraction, they are widely used for almost any case. Also, Neural Networks are able deal with complicated tasks since they are scalable and capable of learning both linear and nonlinear relations.

However, training a neural network is expensive task in terms of processing power and time to tune all the hyperparameters in network. Another most common problem of neural networks is overfitting, which means instead of learning the relation in the training dataset, neural networks could memorize it if they are trained more than enough.

In order to design fully connected neural networks for each T_X, one of the most common framework of Python Programming Language [20] in literature, Keras [21] is used. Single architecture is designed and used for each T_X for the sake of objectiveness. All possible hyperparameters are tuned to make the networks as simple and accurate as possible. For tuning process, all T_X considered equally to find the optimal design.

The main design parameters in neural networks are number of hidden layers and number of nodes in a layer. From the experiments, it is stated that one hidden layer is enough for our task due to low complexity in dataset. Also as demonstrated in Figure 4.2 which shows how mean absolute error changes with respect to number of node in the hidden layer for each optimizer.

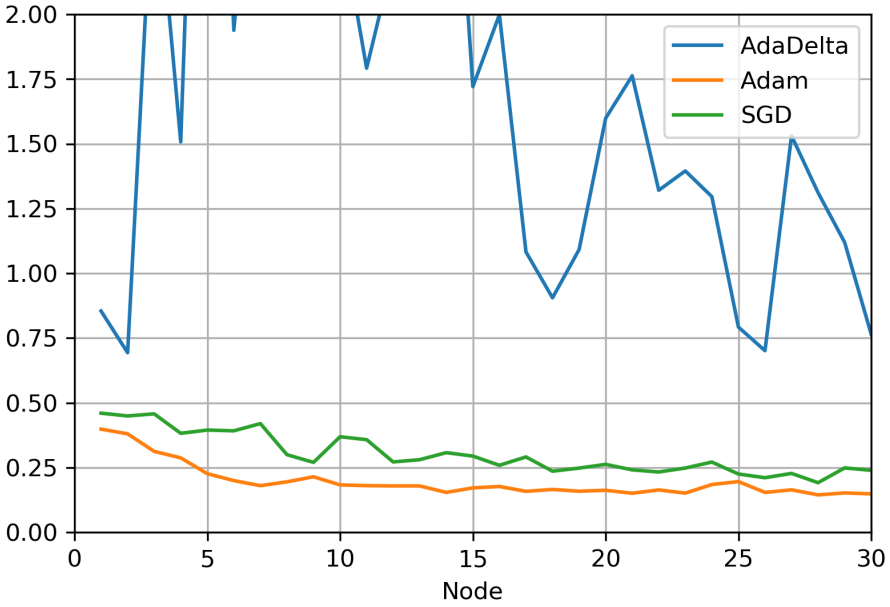

Figure 4.2: System 1 ANN optimization

Activation functions are defined for each layer which corresponds to functionality in hidden node. There are several predefined activation functions such as tanh, sig, relu [22] which are widely used for in literature. For regression tasks, the most popular combination is linear activation output layer for regression and relu [23] for hidden layer in order to extract nonlinear information from dataset. Our experiments are also end up in similar condition where the most accurate results are acquired with this combination. For training procedure, there are numerous different defined algorithms for tuning the parameters for Neural Networks such as Adam [24], Adadelta [25], Stochastic Gradient Decent [26] etc. For regression tasks, Adam is the most suitable and accurate one according to the experiments we have conducted. Besides, the optimization results we obtained are parallel as documented in the literature [27] [28]. To sum up, one hidden layer with 20 nodes and relu activation is optimum with Adam optimizer for training process.

Neural Networks are tend to over-learn the training dataset unless there is a stopping mechanism in the training procedure. To teach the relation within the training dataset instead of memorising, validation and early stopping mechanisms are implemented in training procedure. The training dataset is shuffled and 20% of it used for validation the training algorithms to differentiate the point models start overfitting where the

mean absolute error on validation set started to increase.

In the dataset, there are 11 T_X columns corresponding to 11 independent thermocouple readings. For each T_X, a neural network model with the architecture explained above is trained for many different ν values by modifying the dataset as explained in Section 3.3. For testing the trained networks, test dataset is modified with a fixed ν value according to the Eq 3.1. Most of the ν span is tested by using $\nu = -1$ to $+1$ with the 0.01 step. An example of test input where T_9 is modified by $\nu = 0.5$ to generate noisy T_9' demonstrated in Figure 4.3.

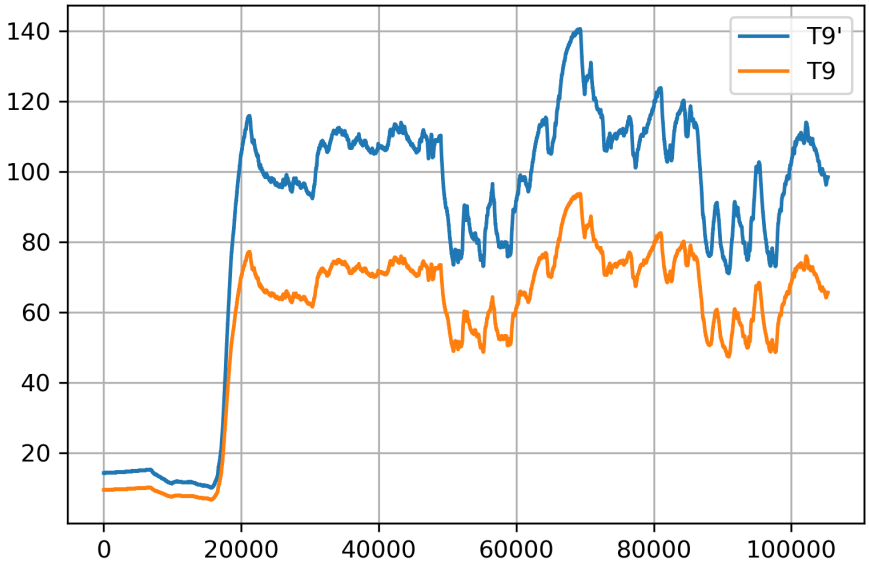

Figure 4.3: T_9 and T_9' (with $\nu = 0.5$).

In order to predict the ν value, the neural network model responsible for the corresponding T_X as trained by training dataset as explained is used.

4.1.2 Model-2 - Decision Tree

Decision Tree is another commonly used supervised predictive model where there is a nonlinear relation in dataset. Although it is known that Decision Tree models are not as capable as other complex models, they are fast and capable of extracting the pattern of data especially for the cases where the dataset is huge. This is why Decision

Tree could be a competitive against Neural Network for our case.

Decision Tree consist of nodes where each node splits the set into two groups according to a logical expression. In the training process, logical expressions in the nodes are tuned in a way to split the dataset as pure as possible. By using most suitable set of logical expression to split the dataset, Decision Trees are capable of extracting the systematic relation in dataset whether it is linear or not.

The most important hyperparameter in a Decision Tree is tree depth which indicates the number of logical split operation in decision tree. As it is valid for Neural Network, higher tree depths could lead to overfitted model where model could not be generalized and usable for any data other than training dataset.

As we did in Neural Network architecture, an optimal Decision Tree structure is built and used for all T_X. We implemented Decision Tree with several different maximum tree depth number and test for all possible noise ratio as its done for Neural Network and found out that 10 is sufficient enough for accurate noise ratio prediction as in Figure 4.4.

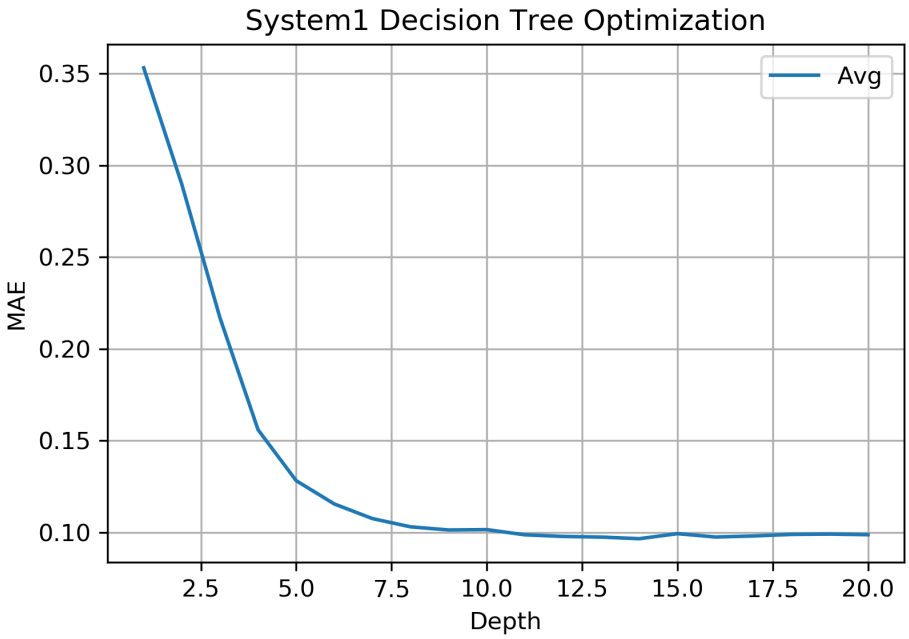

Figure 4.4: System-1 Model-2 Decision Tree optimization.

4.2 System-2 - Data Prediction

We tried to predict the noise ratio of a T'_X from given set of data by using two supervised prediction models; Artificial Neural Network and Decision Tree. Noisy data could lead to misinterpretation of the helicopter during real time data monitoring and cancelling the test unnecessarily. Fortunately, using System-1, it could be possible to understand whether the data is valid or not in order to neglect. Although knowing the data noisy, helicopter will be invisible for analysis without the actual data. It could lead to reconducting the test which would be expensive or more solemnly, make the any possible accident unnoticeable. Therefore, recovering the data could be vital for both real time data monitoring and data analysis.

4.2.1 Model-1 - Linear Model

System-1 able to predict the ν for any T'_X using a supervised predictor model. Since the noise function is known from Section 3.3, T_X could be recalculated from T'_X and predicted ν by using the inverse of noise function as follows;

$$T_X = \frac{T'_X}{1 + \nu} \tag{4.1}$$

Clearly, this approach transfers the error from System-1 directly in prediction since there is only a linear operation. Also, there is no learning procedure or hyperparameter in this approach so that no further testing or optimization is required.

4.2.2 Model-2 - Artificial Neural Network

As a second approach, Artificial Neural Network is used for data prediction. In this case, neural networks are trained to predict the T_X value for a given set of data where T_X is modified to T'_X according to the noise generation principle in Section 3.3. In addition to that, predicted ν by System-1 is also provided to network as an input.

Same design approach in System-1 also applied for this case, 11 independent neural networks are trained for each T_X. Same hyperparameter tuning experiments are also

conducted and as expected similar results are obtained. Linear activation function used output layer and Relu is used for hidden layer for non-linearity. Adam optimizer is used as training algorithm and in order to avoid overfitting, 20% of dataset is used for validation for early stopping mechanism. Different number of nodes in hidden layer are tried and it is stated that 20 nodes with Adam optimizer is sufficient for accurate data prediction as its shown in Figure 4.5.

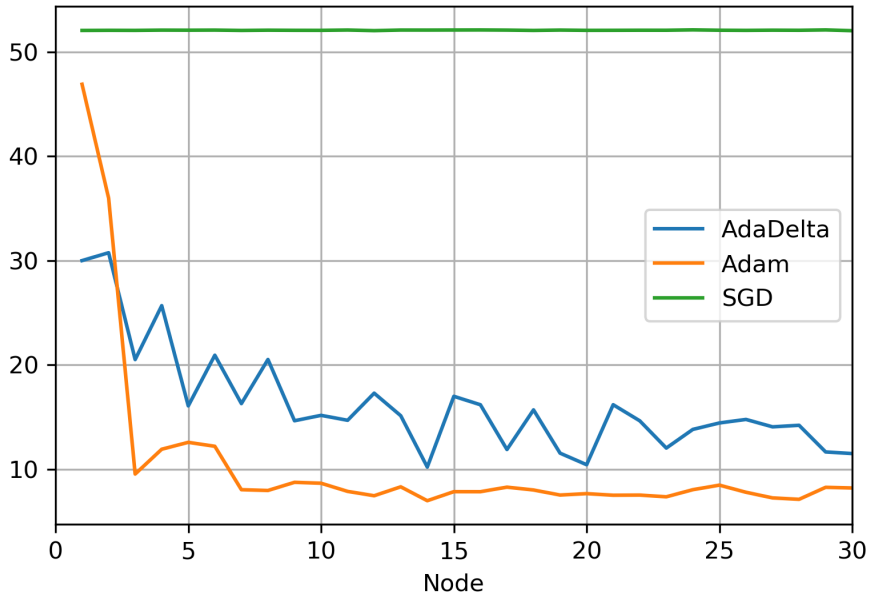

Figure 4.5: System-2 ANN optimization

In Figure 4.5, different from Figure 4.2, Y axis indicated the Mean Absolute Percentage Error because there is no boundary for output in System-2 since the temperature value could be from 15 °C to 120 °C as in can be seen in Figure 3.2 and Mean Absolute Error would be misleading for System-2. However, in System-1, the expected output is in the range of -1 to +1 and Mean Absolute Error is valid since it is forced to be in that range by our Noise Generation Principle defined in Section 3.3.

4.2.3 Model-3 - Decision Tree

As a last model, Decision Tree employed for System-2 as it is done and successful in System-1. We use similar training and testing approach in System-1. From the optimization experiment, it could be seen from the training error pattern in Figure 4.6,

10 depth is the optimal point for Decision Tree algorithm.

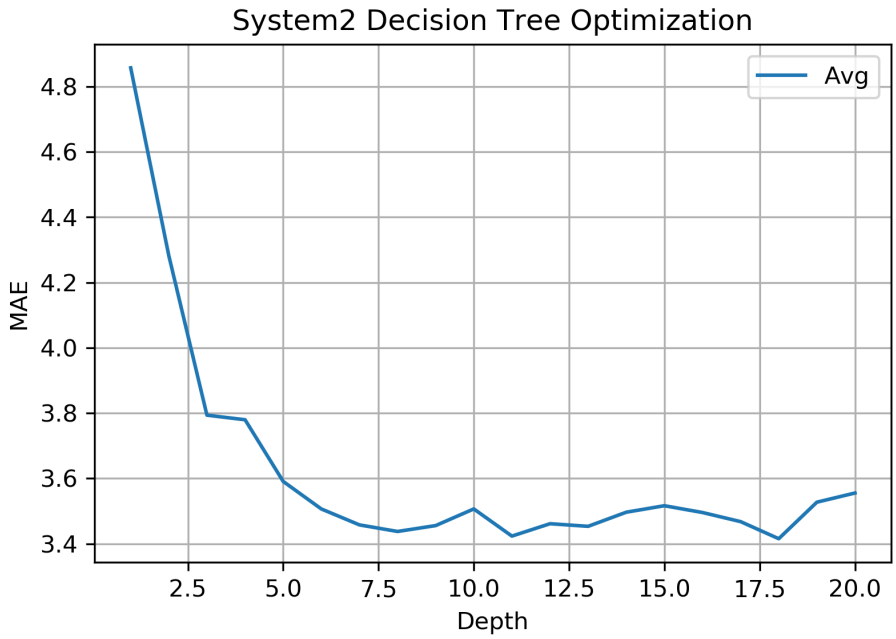

Figure 4.6: System-2 Decision Tree optimization

Decision Tree algorithm tries to classify data by the logical expressions which would also works accurately for the regression tasks if the output has limited output range like in System-1. However, Decision Tree's training error is higher than Artificial Neural Network for System-2 since the output is unlimited.

SECTION 5

RESULTS

In this thesis, a sequential system consist of 2 subsystems that one of them responsible for predicting noise ratio of a T_X from any given set of data and the next system tries to regenerate the actual data from given dataset and predicted noise ratio. In the Section 4, different architecture designs are proposed. These proposed design composed of predictive models are going to be tested with a separate set of data which contains same temperature reading from another flight test of the helicopter with same test configuration on a computer with 16GB RAM, Intel i7-10750H CPU with 2.6Ghz and NVDIA GeForce GTX 1650ti GPU.

In this section, each individual model of two systems are going to be tested in the whole range of expected noise ratio from -1 to $+1$ with 0.1 increment. All 11 T_X are going to be tested one by one in order to evaluate the model's performance in detail. We demonstrate the results with a cross plot where Y axis for corresponding error for each T_X to X axis for Noise Ratio ν. The Y axis of resulting plots are scaled to same level within each system to make the comparison easier.

5.1 System-1

System-1 is responsible for predicting the ν value for any desired T_X from given set of data. The supervised predictive model is proposed: Artificial Neural Network and Decision Tree as explained in Section 4.1. Mean Absolute Error is chosen as Error Metric for System-1 since MAE represents the error accurately for the cases where the error is bounded in a close range like we have ν in the range of -1 to $+1$. Each proposed model is going to be evaluated one by one in following sections.

5.1.1 Model-1

The first proposed model for System-1 was Artificial Neural Network. For each T_X, a separate model with a optimal architecture is trained using training data set as it is explained in Section 4.1.1 in detail. Test set is modified for each T_X by a constant ν value from -1 to $+1$ with 0.1 step size repetitively and compared with the predicted results by the corresponding model from the array of trained models. The calculated MAE vs ν values for each T_X are demonstrated in the cross plot in Figure 5.1.

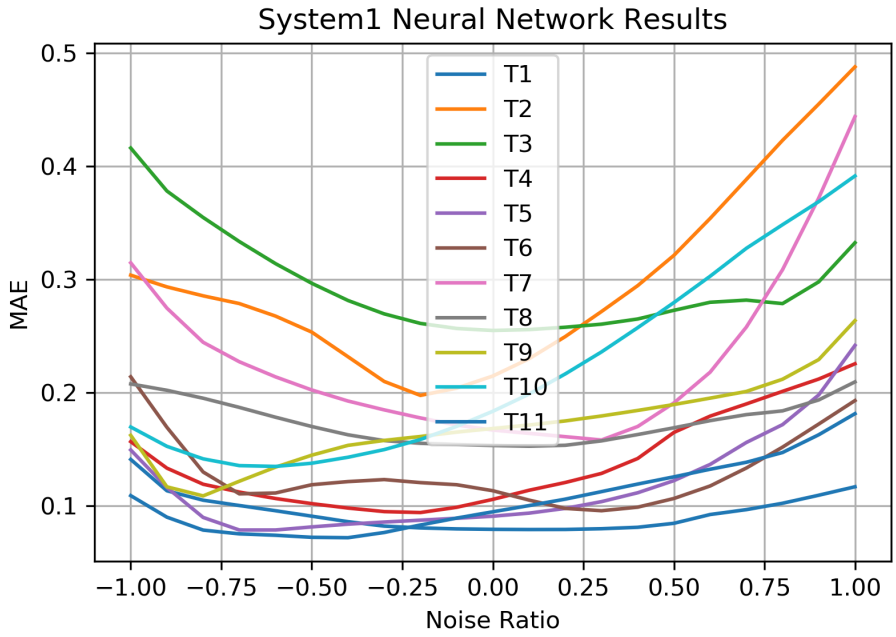

Figure 5.1: System-1 ANN result

It could be seen from Figure 5.1, error curve of all T_X are convex with the valley point at around $\nu = 0$. It states that there is a negative correlation between prediction accuracy and Error Rate ν which is intuitively expected. It could be claimed that any noise ratio could be predicted with and average of 0.15 error although, there are some sensors that could be predicted more difficult than the others such as T_2 and T_3 that could be happened due to the location of corresponding thermocouple on engine. Also, error curve of T_{10} is different from remaining ones which is expected since it's characteristic is different than the other ones as it could be seen in Figure 3.2.

5.1.2 Model-2

The second proposed model for System-1 is Decision Tree algorithm. Same testing approach is applied where all T_X values are modified one by one and corresponding trained model is used for predicting ν value. MAE vs ν cross plot is provided in Figure 5.2.

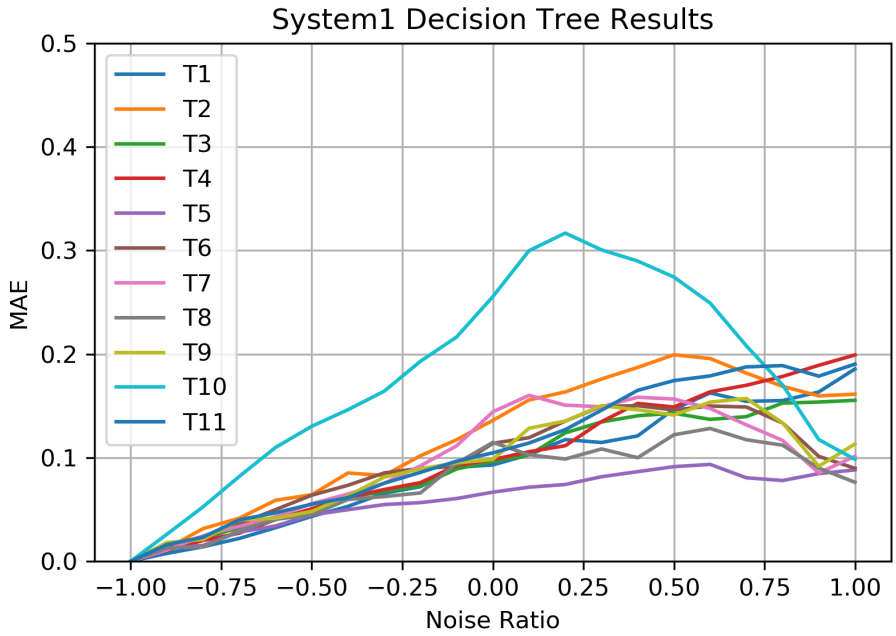

Figure 5.2: System-1 Decision Tree result

The error curves of Decision Tree algorithm is completely different than Neural Network. While Neural Network's error curves are mostly convex shape, Decision Tree algorithm has almost linear graph with a bending around $\nu = 0.5$. Another interesting observation is Decision Tree worked perfectly for the cases where $\nu = -1$ which indicates the success of Decision Tree algorithm's extracting the systematic pattern of dataset as it is explained in Section 4.1.2. $T_X = 0$ is possible only when $\nu = 0$ and small T_X values are possible when ν is closer to -1. This systematic could be easily learned by Decision Tree. Besides, as expected, T_{10} which could be considered as outlier in dataset has different error curve.

Surprisingly, when we compare Model-1 with Model-2, it is seen that Decision Tree algorithm predicts noise ratio more accurately than Artificial Neural Network. Alt-

hough the average MAE for Model-1 is 0.15, Model-2 bests it with 0.1 MAE which makes the Decision Tree algorithm as winner for System-1.

5.2 System-2

System-2 is aimed to predicting the actual T_X from given dataset and ν value predicted by System-1 which is vital and crucial for the flight test scenarios since when a temperature reading is invalid, helicopter become invisible for analysis. Three approach is proposed for System-2 that are explained in following sections in detail. Different from System-1, MAE would be misleading for evaluating the System-2 since the output is no longer bounded. Furthermore, Mean Absolute Percentage Error (MAPE) would be used as Error Metric for testing the proposed models.

5.2.1 Model-1

As a first model, linear transformation is proposed. Since the ν value could be predicted by System-1, it is possible to recalculate the T_X since both T'_X and ν is known according to the inverse of noise function as in Equation 4.1.

We have perfectly accurate System-1 for $\nu = 0$ due to the success of Decision Tree algorithm for classifying the systematic within dataset. However, in the Equation 4.1, when we substitute ν with -1, the equation goes to infinity so that Model-1 of System-2 could not recover the data even though its accurately labeled. Furthermore, for Model-1, the test scenario where $\nu = -1$ is omitted. Also for the following models, the accuracy without $\nu = 0$ is going to be calculated for fair comparison. The average MAPE for Model-1 is calculated to be 11.35%. The detailed results are demonstrated in Figure 5.3

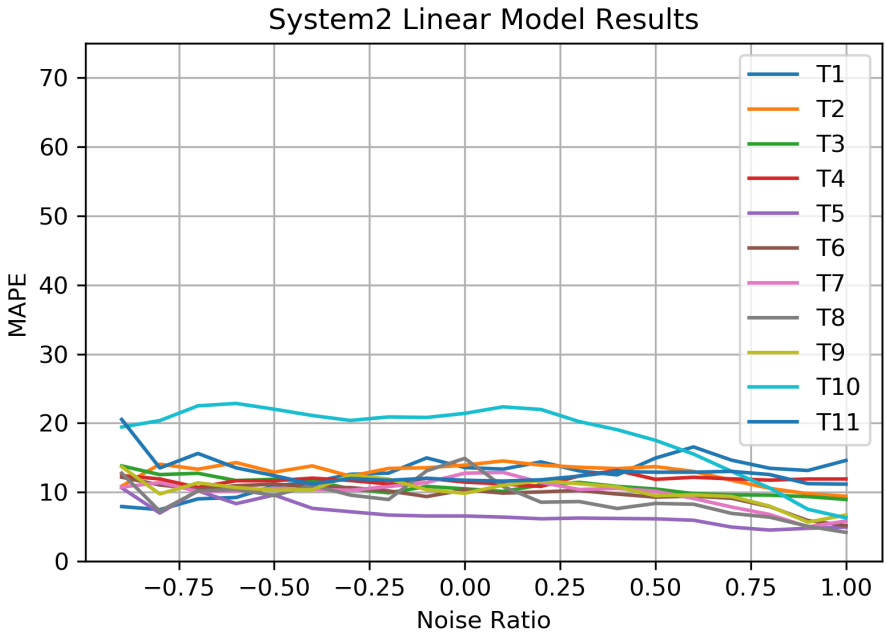

Figure 5.3: System-2 Linear Model results

All the result plots are demonstrated in same Y axis scale for comparison. It could be seen from Figure 5.3, Model-1 could predict T_X values with same accuracy independent from T_X and ν values. However, as expected T_{10} have higher and unrelated curve graph compared to other ones.

5.2.2 Model-2

As an alternative supervised predictive model, our third model is a again Artificial Neural Network. As it is explained in Section 4.2.3, a unique and optimal ANN is trained for each T_X using training dataset and tested for different level of noise ratio values. The average MAPE for Model-3 is computed as 8.28% which is significantly better than other benchmark Linear Model and Decision Tree algorithm. The results for each T_X and ν values are provided in Figure 5.4.

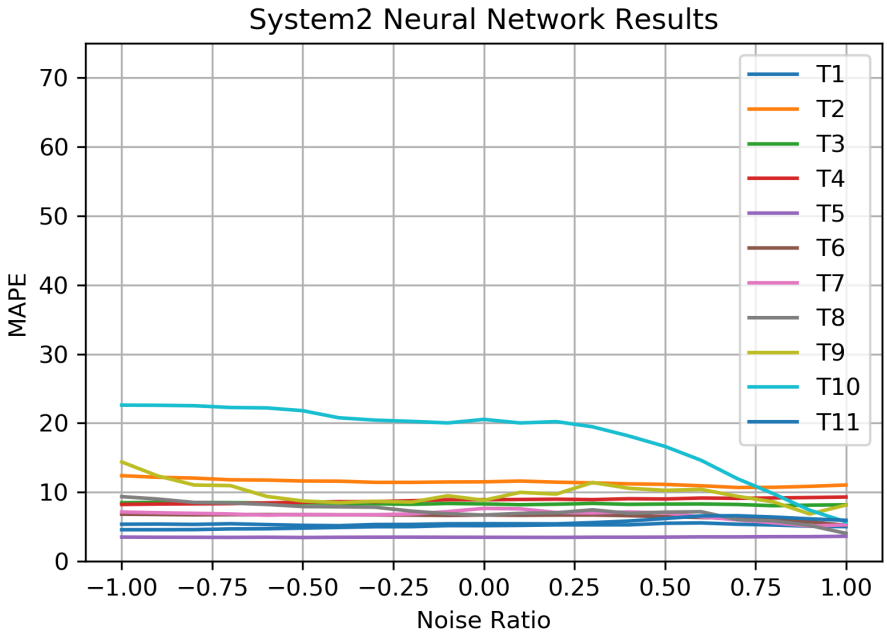

Figure 5.4: System-2 ANN results

Again T_{10} behaves differently from the other T_X since it is considered as outlier in dataset due to corresponding thermocouple's location on engine. However, not only having the lowest error rates among the other models, ANN provides even more stable results than Linear Model which makes the Model-3 the most accurate and reliable proposal for System-2.

5.2.3 Model-3

The second proposed model for System-2 is winner of the System-1, Decision Tree algorithm. The most optimal Decision Tree model was trained on training dataset and tested for different level of ν values for each T_X. The average MAPE for Model-2 is calculated as 19.12% (14.84% without $\nu = 0$) which is surprisingly even worse than simple Linear Model described in Model-1. Decision Tree algorithm failed since it tends to classify the outputs instead of regressing which worked perfectly for System-1. The resulting graph is provided in Figure 5.5.

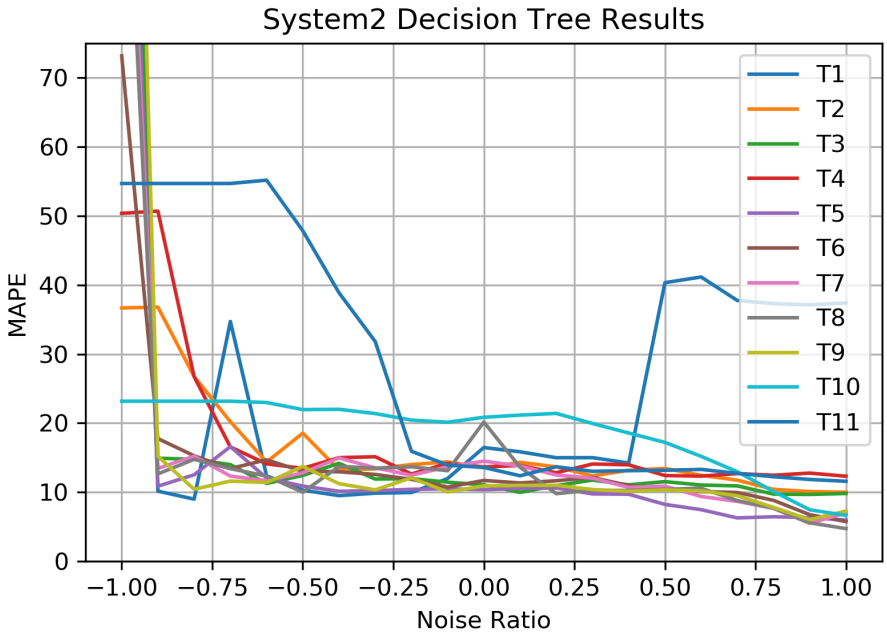

Figure 5.5: System-2 Decision Tree results

As it could be seen from Figure 5.5, In addition to low accuracy of predicting T_X values, Decision Tree algorithm could not provide stable results too. There are unexplainable high error values in T_1 and T_{11} error curves and even more than 100% error rates for $\nu = 0$ which makes the model unreliable.

5.3 Further Testing

Although the system designed with the assumption of single point of failure where only one sensor could be corrupted during test, it would be interesting the analyse the proposed system's performance if two or more sensor readings are invalid. In order to test this scenario, two sensor readings are corrupted with uniform noise distribution for each observation. This synthetically manipulated dataset is used for testing the models for predicting one of the corrupted reading. The detailed results for Noise Prediction and Data Prediction are shown in Figure 5.6 and Figure 5.7.

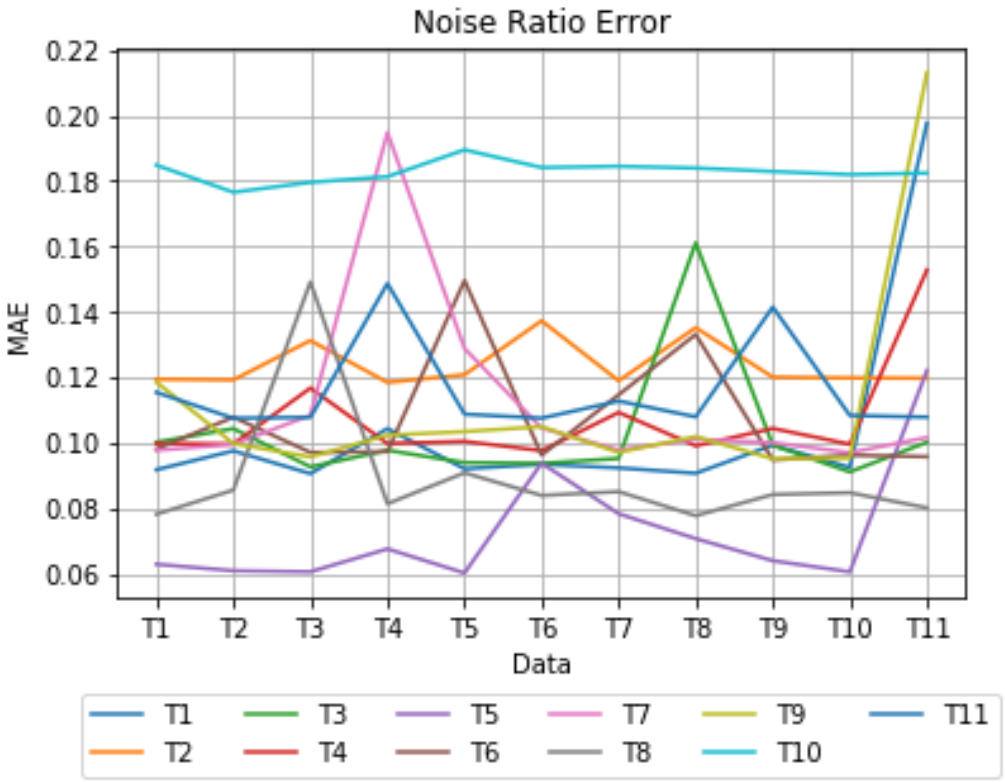

Figure 5.6: Noise prediction results for double sensor failure

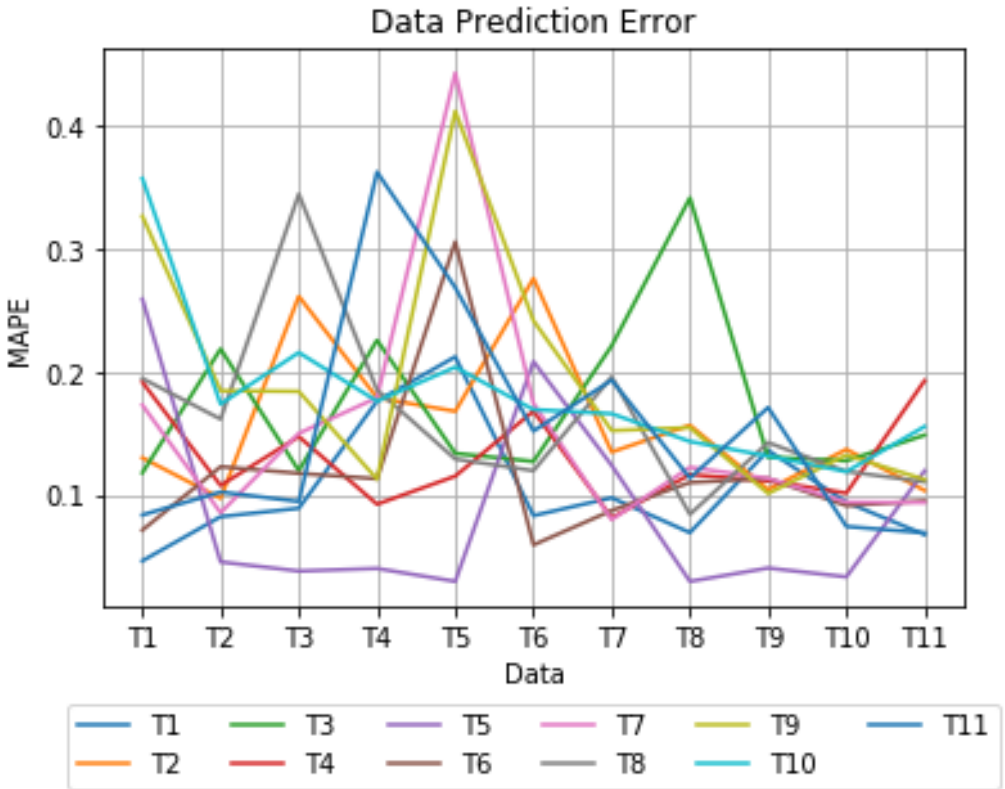

Figure 5.7: Data prediction results for double sensor failure

Before the testing Double Sensor Failure case in each System, corresponding T_X is corrupted with ν value for with uniform distribution between -1 to $+1$ according to Equation 3.1 for each data observation. In addition to the aimed T_X reading, another reading from dataset is corrupted with same principle which defined in X axis and MAE for ν prediction or MAPE for data prediction is calculated as in Y axis in Figure 5.6 and Figure 5.7. For example, in Figure 5.6, the pink line indicates the MAE of ν prediction for T_7. The intersection point of $X = T_4$ and the pink line indicates the MAE of ν prediction for T_7 in the scenario where both T_7 and T_4 are corrupted with uniform random ν value.

The error rates are mostly stable expect for one or two spikes for different T_X values. These spikes are occurred in different data index for each T_X reading which shows the correlation of T_X reading in dataset. For example, T_7 is highly correlated with T_4 so that the noise ratio error for ν prediction for T_7 increased tremendously when T_4 is corrupted in Figure 5.6. Also in Figure 5.7, it could be seen that T_3 is highly correlated with T_8 in data prediction case.

It is also interesting to investigate the performance of the system in multiple sensor failure cases. In order to test this phenomena, targeted T_X is synthetically corrupted with uniform ν value and other sensor readings are manipulated one by one in order. The results for System-1 and System-2 are demonstrated in Figure 5.8 and Figure 5.9

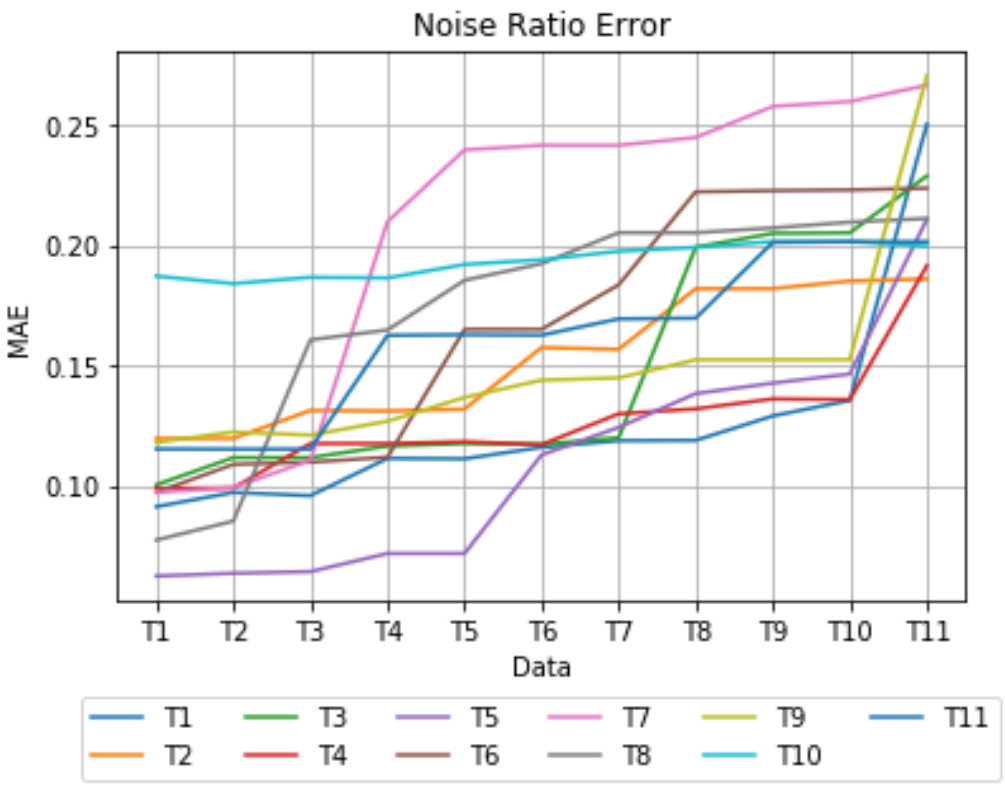

Figure 5.8: Noise prediction results for multiple sensor failure

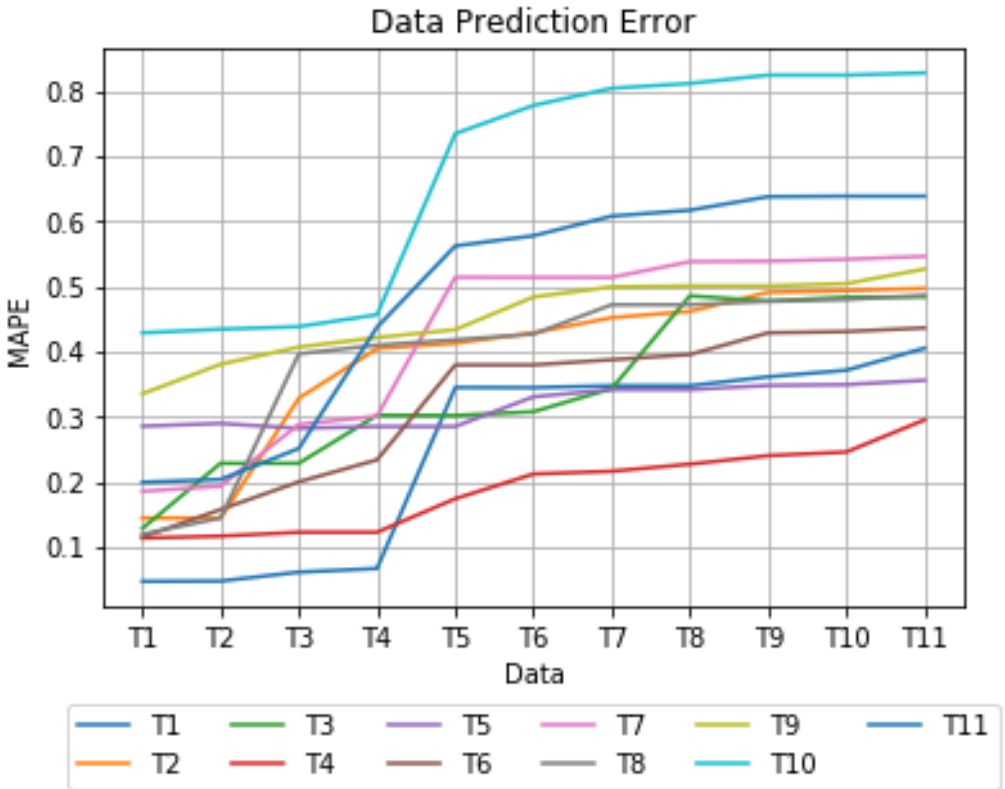

Figure 5.9: Data prediction results for multiple sensor failure

In Figure 5.8 and Figure 5.9, corresponding T_X measurement is corrupted in the beginning and as it is labeled in X axis, additionally one more T_X is corrupted in $+X$ direction.For example, in Figure 5.8, the pink line indicates the MAE of ν prediction for T_7. The intersection point of $X = T_4$ and the pink line indicates the MAE of ν prediction for T_7 in the scenario where T_7 and $[T_1, T_2, T_3, T_4]$ are corrupted with uniform random ν value.

As expected, noise ratio value increases as we increase the number of corrupted measurements. Also the correlation between measurements could be realized in higher increments in noise ratio values as it is discussed in Double Point of Failure tests. As it could be seen from Figure 5.8 and Figure 5.9, T_7 is highly correlated with T_4 in Noise Ratio prediction so that when T_4 is corrupted, the accumulated error increases massively. In addition, for Data Prediction, accumulated error rate for T_3 jumps when T_8 is corrupted.

5.4 Scenario Testing

During an actual flight test, it is more common to have a constant noise on a measurement for a period of time due to external affects. In order to test the proposed system in a more realistic way, 6 randomly picked different Noise Ratio values $-0.75, -0.5, -0.25, 0.25, 0.5, 0.75$ applied on a 6 randomly selected measurement for short period of time between $40000^{th} - 43000^{th}$ observations which is demonstrated in Figure 5.10. In Figures 5.11 to 5.16, the predicted result for these scenarios are shown where corresponding Noise Ratio applied on a selected measurement.

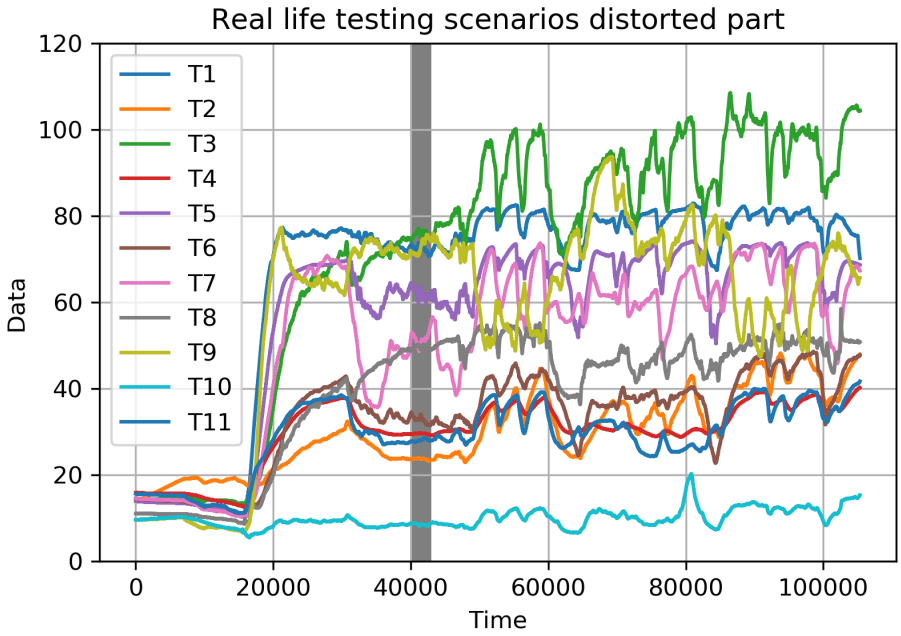

Figure 5.10: Distorted portion of data in real life testing scenarios

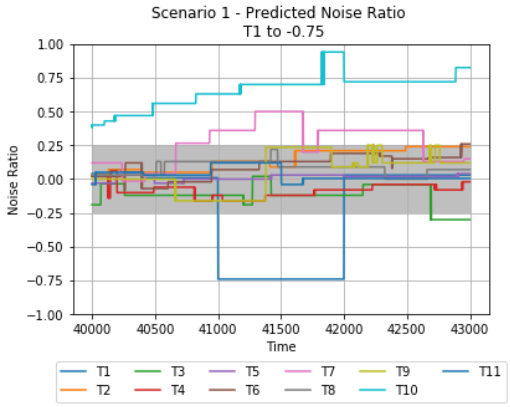

Figure 5.11: Real life testing scenario 1, -0.75 ν on T_1

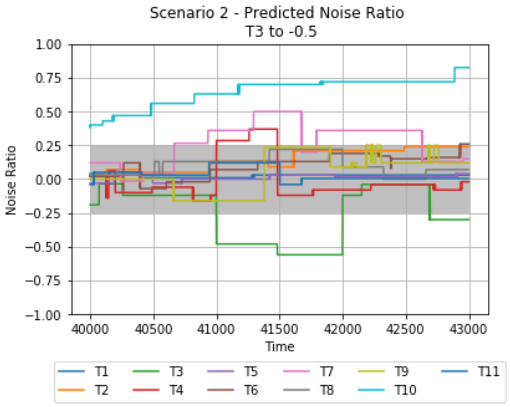

Figure 5.12: Real life testing scenario 2, -0.5 ν on T_3

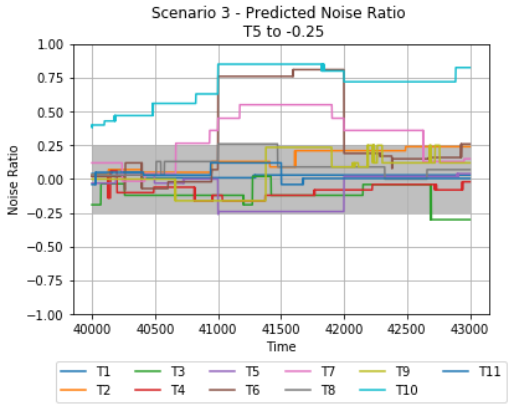

Figure 5.13: Real life testing scenario 3, -0.25 ν on T_5

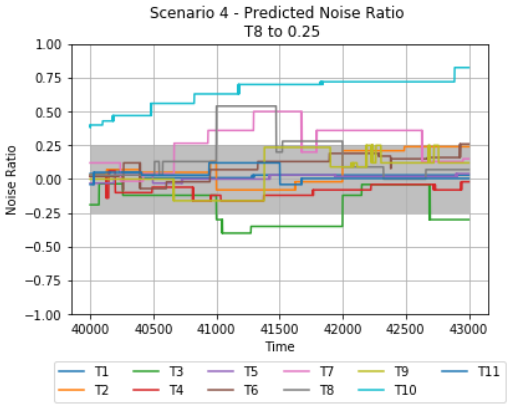

Figure 5.14: Real life testing scenario 4, 0.25 ν on T_8

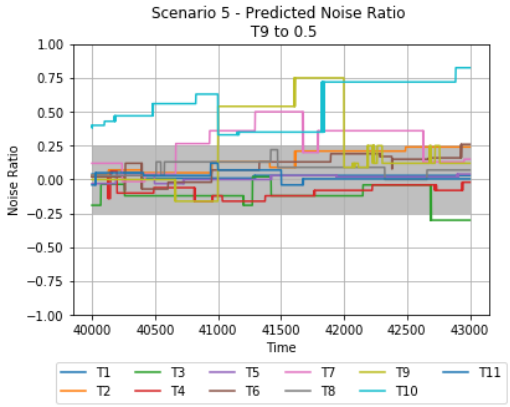

Figure 5.15: Real life testing scenario 5, 0.5 ν on T_9

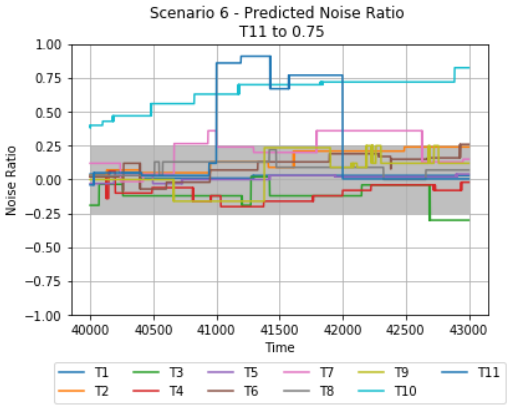

Figure 5.16: Real life testing scenario 6, 0.75 ν on T_11

As it could be seen from the plots above, System-1 predicts between ±0.25 Noise Ratio for a $\nu = 0$ measurements which is called acceptable range that is colored like gray zone in plots. For the observations between 41000^{th} – 42000^{th} where a measurement is corrupted with a given ν, predicted results are not in acceptable range. Even for $\nu = \pm 0.75$ and ± 0.50, the system could easily identify the noisy measurement. However when $\nu = \pm 0.25$ where the applied noise ratio within the acceptable results, the system predicts some noise ratio values outside of range but could not detect the noisy measurement successfully. As we realized for all testing procedures, T_10 is an outlier in our dataset that could be ignored.

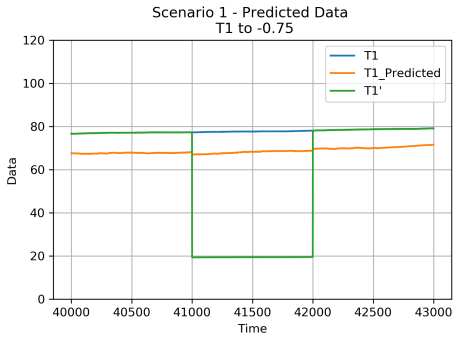

Figure 5.17: Data prediction prediction in real life testing scenario 1

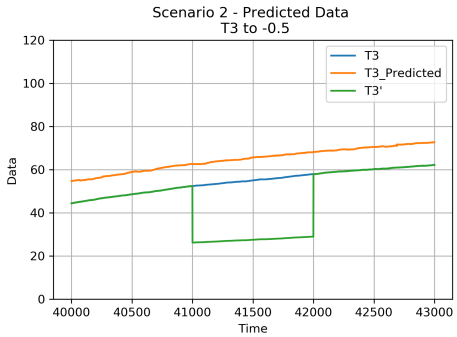

Figure 5.18: Data prediction prediction in real life testing scenario 2

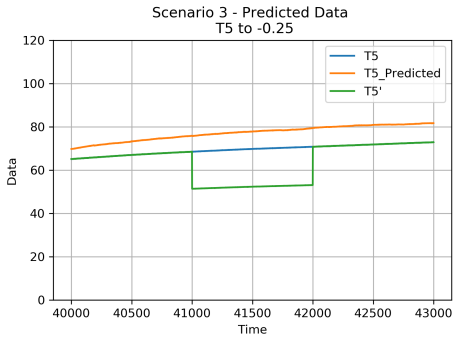

Figure 5.19: Data prediction prediction in real life testing scenario 3

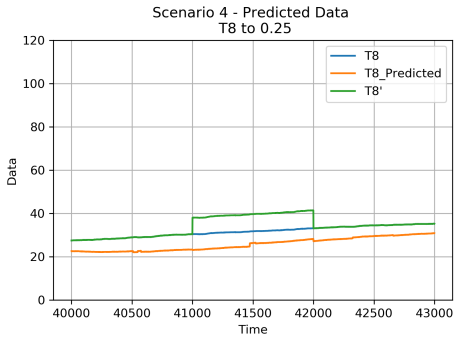

Figure 5.20: Data prediction prediction in real life testing scenario 4

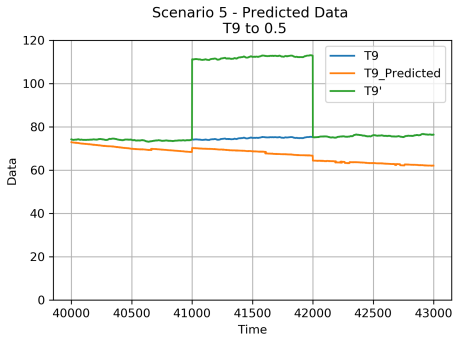

Figure 5.21: Data prediction prediction in real life testing scenario 5

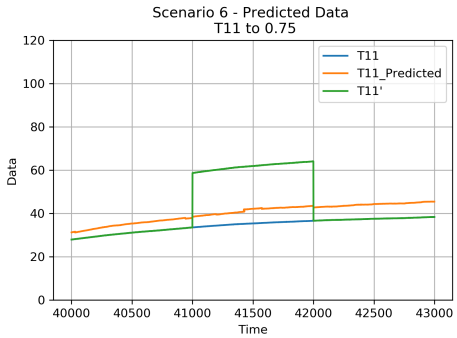

Figure 5.22: Data prediction prediction in real life testing scenario 6

For real life testing scenarios, as it could be seen from Figure 5.17 to 5.22, System-2 could predict the actual value with approximately 0.08 MAE as it is calculated independent from the noise ratio applied on actual measurement.

5.5 Analysis

As it is explained in detail in previous sections, the architecture we proposed could predict the amount of noise on any selected measurement with 0.1 MAE by System-1 using DT model and predict the actual measurement with 0.08 MAPE by System-2 using ANN. However, around $\pm 0.25\nu$ which is called gray zone, the system is not accurate due to the error in System-1. Although there is no noise on measurement, the system predicts a ν value in gray zone. Also for the cases where measurement have noise value in gray zone, system detects that there is something unexpected in data although it could not identify the noisy measurement. Also as it is explained in Dataset section in detail, T_{10} acts as an outlier since there is no correlation between other measurements in dataset due to its location on engine. So in order to train the models properly, measurements shall be selected by their physical relation on helicopter.

Also the architecture is tested on more harsher environment than its trained. The systems reaction on the scenarios where two or more sensor is corrupted is demonstrated although system is not trained for these cases. On these scenarios system could be able to reproduce data for some extent even though its not trained for such cases.

5.6 Future Work

The selected dataset is includes the temperature measurements from an engine of helicopter. For proof of concept, the architecture is trained for this subset of whole measurements in helicopter. However, there are other subsystems in helicopters such as structure, rotor, transmission, avionics etc. and for each of these subsystems, there are different kind of measurements such as vibration, strain, force, displacement etc. For each of these subsystem and also for each type of measurement, measurements from helicopter shall be grouped by their physical relation and correlation and individual systems shall be trained for each group in order to cover all measurements from helicopter.

The FTI measurements from helicopter is time series data through the flight test of helicopter. ANN and DT are the proposed models for each system. However, these

predictive models are trained and designed in a way that there is no time relation in measurement. Instead of using this kind of models, more complex models that also extracts the time relation such as Long Short Term Memory (LSTM), Convolutional Neural Network (CNN) could be used which could achieve higher accuracy.

The proposed architecture includes predictive models with most optimize and simple architecture as possible. Also each model for each measurement could work independent from others. During the real time data monitoring in actual flight test, the validity of critical measurements could be predicted by using System-1. From the predicted v values from System-1, corresponding models from System-2 could be used for data prediction for measurements in interest. So that, instead of using whole models during real time monitoring, it is possible to select the measurements to process which would decrease the processing time tremendously.

SECTION 6

CONCLUSION

Helicopters are one of the most commonly used aircraft for both civilian and military purposes due to their high maneuver capabilities. However, designing a helicopter from scratch is a challenging process since there are lots of dynamic and electronic subsystem that shall run in harmony. Although all possible simulations are conducted in digital environment, actual testing of a final product is unavoidable. Especially actual flight tests of newly produced prototypes are dangerous so that helicopters health status are monitored in real time as much as possible. In order to track any physical or electronic event during flight test, helicopter is instrumented with analog sensors using Flight Test Instrumentation Data Acquisition Units which are also capable of gathering electronic communication on helicopter itself. The data collected with FTI system are transmitted to a ground station using telemetry and expert fields are responsible for interpretation of data in order to prevent any possible accidents. Unfortunately, analog sensors could be easily damaged or miscalibrated on prototype helicopters since they are generally on the surface and vulnerable to external effects. Damaged sensors produce meaningless data which could end up serious consequences such as repeating the test which is enormous time and money waste or even unforesee upcoming accident. In order to calculate the amount of noise on data and regenerate the actual data from it, we trained supervised predictive models such as Decision Tree and Artificial Neural Networks using actual temperature data from an engine of helicopter during actual flight test. Decision Tree model could predict the noise level with 0.1 MAE, whereas Artificial Neural Network could regenerate the actual data with 8.28 MAPE. Furthermore, we propose a hybrid sequential system with a subsystem that could predict noise ratio with Decision Tree algorithm and Ar-

tificial Neural Networks is able to reproduce the actual data. This hybrid model could be used in real time to decide the confidence level of data and reproduced actual data could be used for flight test data analysis without repeating the flight test saving time and finances in testing process.

Acknowledgment

This research is conducted and completed using the flight test data provided by the Turkish Aerospace. Neither the data itself nor the helicopter for which the data is collected could be revealed due to security and confidentiality issues.

References

[1] W. Johnson, *Helicopter Theory*, ser. Dover Books on Aeronautical Engineering Series. Dover Publications, 1994, ISBN: 9780486682303. [Online]. Available: https://books.google.com.tr/books?id=SgZheyNeXJIC.

[2] G. Done, "Helicopter Flight Dynamics: the Theory and Application of Flying Qualities and Simulation Modelling – Second edition G.D. Padfield Blackwell Publishing, 9600 Garsington Road, Oxford, OX4 2DQ, UK. 2007. 641pp, Illustrated. £129. ISBN 978-1-4051-1817-0.", *The Aeronautical Journal*, vol. 113, no. 1141, p. 202, 2009. DOI: 10.1017/s0001924000087959.

[3] A. F. T. P. S. E. A. Ca, "Volume IV. Flight Test Management. Chapter 5. Flight Test Instrumentation.", 1990. DOI: 10.21236/ada320062.

[4] Secretariat, R. C. Council, White, and S. M. Range, "Telemetry Standards, IRIG Standard 106", 2007.

[5] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.", *Psychological Review*, vol. 65, no. 6, 386–408, 1958. DOI: 10.1037/h0042519.

[6] J. V. Ryzin, L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees.", *Journal of the American Statistical Association*, vol. 81, no. 393, p. 253, 1986. DOI: 10.2307/2288003.

[7] T.-H. Guo and J. Nurre, "Sensor Failure Detection and Recovery by Neural Networks", *International Joint Conference on Neural Networks*, 1991.

[8] E. Ezhilrajan, S. Rajapandian, and L. L. S. Titus, "An Intelligence System Model to Study the Impact of Protruding length on Temperature Sensors for Cryogenic Propulsion", *2020 International Conference on Computer, Electrical Communication Engineering (ICCECE)*, 2020. DOI: 10.1109/iccece48148.2020.9223084.

[9] S. O. Ogaji and R. Singh, "Advanced engine diagnostics using artificial neural networks", *Applied soft computing*, vol. 3, no. 3, pp. 259–271, 2003.

[10] M. R. Napolitano, D. A. Windon, J. L. Casanova, M. Innocenti, and G. Silvestri, "Kalman filters and neural-network schemes for sensor validation in flight control systems", *IEEE Transactions on Control Systems Technology*, vol. 6, no. 5, pp. 596–611, 1998. DOI: 10.1109/87.709495.

[11] H. Huang, J. L. Vian, J. Choi, D. Carlson, and D. C. W. Ii, "Neural network inverse models for propulsion vibration diagnostics", *Applications and Science of Computational Intelligence IV*, 2001. DOI: 10.1117/12.421182.

[12] R. Yang, P. V. Er, Z. Wang, and K. K. Tan, "An RBF neural network approach towards precision motion system with selective sensor fusion", *Neurocomputing*, vol. 199, pp. 31–39, 2016. DOI: 10.1016/j.neucom.2016.01.093.

[13] E. Mese and D. A. Torrey, "An approach for sensorless position estimation for switched reluctance motors using artifical neural networks", *IEEE Transactions on Power Electronics*, vol. 17, no. 1, pp. 66–75, 2002.

[14] C. Alippi, S. Ntalampiras, and M. Roveri, "A Cognitive Fault Diagnosis System for Distributed Sensor Networks", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 8, pp. 1213–1226, 2013. DOI: 10.1109/TNNLS.2013.2253491.

[15] Y. Liao, E. Hashemi, T. Wang, and B. Yang, "A Learning-Aided Generic Framework for Fault Detection and Recovery of Inertial Sensors in Automated Driving Systems", *IEEE Systems Journal*, pp. 1–11, 2020. DOI: 10.1109/JSYST.2020.3004805.

[16] D. D. Pollock, *Thermocouples: theory and properties*. CRC Press, 2000.

[17] S. Kasap, *Thermoelectric Effects in Metals: Thermocouples*, 2001. [Online]. Available: http://courses.nus.edu.sg/course/elengv/ee3406/seebeck\%20and\%20thermocouple.pdf.

[18] P. Jarry and J. N. Beneat, "Nyquist Criteria", *Digital Communications*, 55–69, 2015. DOI: 10.1016/b978-1-78548-037-9.50005-3.

[19] E. Elnahrawy and B. Nath, "Cleaning and querying noisy sensors", *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications - WSNA '03*, 2003. DOI: 10.1145/941350.941362.

[20] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009, ISBN: 1441412697.

[21] F. Chollet *et al.* (2015). "Keras", [Online]. Available: https://github.com/fchollet/keras.

[22] C. Bircanoğlu and N. Arica, "A comparison of activation functions in artificial neural networks", May 2018, pp. 1–4. DOI: 10.1109/SIU.2018.8404724.

[23] A. F. Agarap, *Deep Learning using Rectified Linear Units (ReLU)*, 2018. [Online]. Available: http://arxiv.org/abs/1803.08375.

[24] D. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*, 2014. [Online]. Available: https://arxiv.org/abs/1412.6980v1.

[25] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method", *CoRR*, vol. abs/1212.5701, 2012. arXiv: 1212.5701. [Online]. Available: http://arxiv.org/abs/1212.5701.

[26] S. Ruder, "An overview of gradient descent optimization algorithms", *arXiv preprint arXiv:1609.04747*, 2016.

[27] K. Swingler, *Applying neural networks: a practical guide*. Kaufman, 2001.

[28] M. J. A. Berry and G. Linoff, *Data mining techniques: for marketing, sales, and customer relationship management*. Wiley, 2011.