PREDICTION OF HEAT EXCHANGER THERMAL FATIGUE FAILURE USING MACHINE LEARNING

Michael Stebbins*

West Texas A&M University

Abstract

The thermally dynamic processes that occur in heat exchangers can lead to thermal fatigue of components. Thermal fatigue is the repeated application of stress to a material through thermal expansion that contributes to crack growth and ultimately material fracture. The tools available for fatigue prediction include best-guess type models, which do not account for the stochastic properties of materials, and laboratory testing, which can provide data for specific components at the expense of lab and process shutdown time. This experiment explores the viability of a new tool in the field of thermal fatigue life prediction that utilizes tools from machine learning to detect complex patterns within simple signal data through the application of pattern finding algorithms that are tuned through supervised training. The presented work below, called the Machine Learning Thermal Fatigue Analysis Tool (ML-TFLAT) first collects and preprocesses data into training and testing sets, then trains an LSTM neural network to detect patterns in the data, and finally the network is tested on new data before it is applied to other systems. This work provides evidence for the utility of the ML-TFLAT described in this work. While this work relies on representative data, testing and statistical analysis of the ML-TFLAT shows that the tool's ability to predict failure shows the potential of this novel tool.

1. Introduction

Fatigue, which is a common fault found in industry, can be a long and slow process that rarely shows noticeable signs before failure. [8] Therefore, tools to predict fatigue failure allow for the execution of predictive maintenance in both industrial and commercial settings. Stochastic modeling tools currently available in the market today for analyzing fatigue systems such as the stress- and strain-life models provide only approximations in their analysis, serving as guidelines more than direct assessment of a component. [17] Because fatigue is a stochastic process, which depends primarily on the individualized aspects of each component, two items can be made in the same way, with the same processes and even from the

*Corresponding author.

Email address: m.stebbins1120@gmail.com (Michael Stebbins)

same stock while exhibiting different fatigue characteristics, solely based on the independent molecular structure of the component's material. [13]

Direct-observation methods for crack propagation analysis such as eddy current and ultrasonic testing exist. [14–16] These methods are significantly more intensive and often require the component to be taken out of service and sent to a lab for analysis, while providing only a snapshot of the system and therefore need to be repeatedly executed to determine a damage rate. The cost of implementation and production down time can easily outweigh the benefits of this more precise data. [18]

This research explores a potentially more individualized tool for predicting thermal fatigue failure by using a less intrusive method that would not require taking equipment out of service for analysis. This method employs the rapidly developing field of machine learning to provide a more accurate prediction of remaining life than the equation-based analysis methods, and it provides this prediction at a much lower cost of implementation and operation than the direct-observation methods. While X. Li et al. [7] , and F. Ahmadzadeh and J. Lundberg [11] have looked into machine learning as a tool, this research proposes a different structure and method for predicting thermal fatigue failures. This Machine Learning Thermal Fatigue Life Analysis Tool (ML-TFLAT) can, in theory, be applied to any system undergoing thermal cycling to analyze the system's thermal fatigue life. With the intended application of this research being the industrial sector, this novel system will be examined using the case of a tube inside a shell-and-tube heat exchanger.

1.1. Description of Experimental Setup

The proposed system is designed to require the fewest amount of field modifications to reduce the installation and operational overhead costs, which allows the data to be collected from a minimal number of inputs. The selected inputs and sources are listed in Table 1:

Table 1: Selected Data Sources for the Heat Exchanger Analysis

Input	**Data Source**
Principal Strains	Bi-axial strain gauge
Pipe Inlet Flow Temperature	Temperature Sensor
Pipe Outlet Flow Temperature	Temperature Sensor
Pipe Wall Thickness	Engineering Drawings/Vendor Manuals
Pipe Flow Pressure	Pressure Sensor
Pipe Flow Rate	Flow Rate Sensor

Most industrial processes have methods to collect the data listed in Table 1 with the exception of principal strains, which would require the installation of a bi-axial strain gauge on the heat exchanger tube being analyzed. [19–21] Since the analyzed system only needs to be interrupted once for the installation of the machine learning thermal fatigue life analysis tool, this method would prove to be more efficient than the direct-observation methods by limiting the process downtime at the minimum. Since the system takes advantage of the

common data sources listed in Table 1 that are likely to be in place already in the facility, the cost of implementation is, most likely, negligible.

Data is collected from the system at regular intervals, summarized at the high and low temperature states of the equipment's process cycle, which provides the amplitude of the strains associated with each thermal cycle. This cycle-scale data is then used to create a time series set of data that can be analyzed by the machine learning algorithm (MLA) of the ML-TFLAT and provides a prediction for the remaining life of the equipment.

For an MLA to be able to make good predictions, it must be trained on a set of data. The training data for the ML-TFLAT would come from numerical models, sourced either from available models (such as those available via digital twin technology) or created explicitly for the system in-house or by contract work. These models provide a wide range of data representative of the system being analyzed. This is done to prevent extensive experimental setup and data collection, which is an unnecessary expense in both time and financial cost when analyzing multiple different types of systems. For the case study in this experiment, the model consists of a thermofluidic model of a pipe with an internal and external flow.

1.2. Machine Learning Process

For this experiment, the numerical model for the system will provide both the training data needed for the MLA and simulated data to evaluate the ML-TFLAT. The data sets used for training will be distinct from the evaluation sets to prevent biases such as over fitting in the data. The process for this experiment is as follows:

1. Generate training data using the numerical model.
2. Train a selection of MLA settings to provide multiple analysis points.
3. Generate evaluation data on parameters that the MLAs have not seen in training.
4. Evaluate the MLAs on the new data to produce statistics describing the performance of each MLA.
5. Evaluate the MLA statistics to determine the usefulness of the tool based on selected settings.

These steps will be explained in more detail in the following sections.

2. Theory and Calculation

2.1. Development of the Thermal Fatigue Model (TFM)

The Thermal Fatigue Model (TFM) presented below uses a finite element approach to determine fatigue damage that operates using thermal cycles as the timestep, allowing the model to be applied to multiple systems simultaneously. A cycle, as defined for this model, starts at the low point temperature for the pipe (typically ambient temperature), increases to the high operating temperature, then returns to the low temperature. Dwell effects and ramp up/ramp down times are not considered and temperatures are assumed to be steady state before advancing through the defined model cycle.

These factors are determined first by a convection/conduction heat transfer system. It then uses the resultant temperature gradient through the material to calculate thermal expansion based on solid mechanics, followed by a calculation of the damage accumulation

using fatigue analysis tools. The method presented here is a novel method based on theory. Work is needed to prove this model is valid, which would include taking progressive samples and observations of crack propagation in a physical setup.

2.1.1. Assumptions Applied to Analysis

Materials. This experiment assumes the material properties of 6061-T6 aluminum for the pipe material and distilled water for the fluid flow. The materials in the model are assumed to be flawless, which removes the effects of corrosion and the stochastic nature of materials such as surface finish and molecular flaws. While this assumption simplifies the model, it reduces the accuracy of the prediction for use in real applications. This is considered acceptable as this experiment is an initial exploration into the usefulness of machine learning in thermal fatigue prediction. A more robust model would be needed for production of a live application.

One-Dimensional Radial Heat Transfer. As the material is assumed to be perfect, the heat transfer is then assumed to be one-dimensional to remove effects of circumferential and axial temperature distributions. This reduces the complexity of the model but can influence the outcome based on the heat transfer characteristics of a real system. This is considered an acceptable assumption as the temperature gradient is likely to be maximum at a single point with a considerably symmetrical distribution in a real system, so analyzing along that one-dimensional direction provides a reliable result.

Flow Regime. The model is designed to handle both turbulent and laminar flow within the pipe, but transitional flow is assumed turbulent for the correlations used. This assumption provides a means to handle cases where the flow falls into the transitional region. With few tools to work with, and by opting to associate the transitional flow regime with turbulent flow, the model conservatively estimates the heat transfer and allows the TFM to produce more conservative predictions than if the flow was assumed to be laminar.

Element Count. For this model, the number of elements to be analyzed between the inner and outer wall of the pipe was chosen to be 61, allowing both a fine-detail approach to the model while enabling the TFM to produce data within a reasonable timeframe. Qualifying information on why the use of 61 elements was chosen can be found in the Model Convergence discussion in of the Results section.

High Cycle Fatigue. As many industrial applications typically are designed for longevity, this model assumes a high-cycle fatigue case where the number of cycles to failure is measured in the thousands of cycles or greater. [4] This best represents the environment in which the ML-TFLAT was designed to be used.

2.1.2. Methods Used in Calculation of the TFM

The core workflow of the TFM is split into four calculation segments: temperature gradient, thermal and pressure stresses, accumulation of damage, and failure assessment.

Temperature Gradient. The first step in thermal fatigue analysis is to calculate the temperature gradient in the material. The TFM model uses the geometry of the pipe, working fluid properties and inlet and outlet temperatures of the section being modeled to determine the convective heat transfer to the cylinder. To find the convective heat transfer coefficient h, the Nusselt number is needed, which can be determined per Subramanian [9] using the Gnielsinski correlation, which is shown to be valid for Reynolds numbers between $4*10^3$ and $1*10^6$ and is determined using Equation (1) , or the Sieder-Tate Correlation for Reynolds numbers less than 2000, shown in Equation (2) :

$$Nu = \frac{\left(\left(\frac{f}{2}\right)(Re - 1000)\,Pr\right)}{1 + 12.7\left(Pr^{\left(\frac{2}{3}\right)} - 1\right)\sqrt{\frac{f}{2}}}\;;\; 4 * 10^3 < Re < 10^6 \tag{1}$$

$$Nu = 3.66 + \frac{0.065\;RePr\left(\frac{D}{L}\right)}{1 \pm 0.04\left(RePr\left(\frac{D}{L}\right)\right)^{\frac{2}{3}}}; Re < 2000 \tag{2}$$

For the above equations, Re is the Reynolds number and Pr is the Prandtl number for the fluid flow.

Reynolds numbers that fall between 2000 and 4000 are considered transitional flow and are difficult to characterize, so the model does not account for those flows and instead assumes the flow is either laminar or turbulent. The convective heat transfer coefficient is then found using the definition of the Nusselt number per Bergman et al. [12] , given in Equation (3) .

$$Nu = \frac{hL}{k} => h = \frac{kNu}{L} \tag{3}$$

For the above equations, h is the convective heat transfer coefficient, k is the conductive heat transfer coefficient, and L is the cylinder length.

The convective heat transfer coefficient is then used with the thermal energy equation for convection in a cylinder per Bergman et al. [12] given by Equation (4) :

$$q = h\,(\pi DL)\,(T_2 - T_1) \tag{4}$$

For the above equations: D is the hydraulic diameter, L is the cylinder length, T_1 is the inner surface temperature, and T_2 is the outer surface temperature.

It should be noted that the convection coefficient here is averaged across the length of the model, so it is advised to use a shorter modeling length for a more accurate heat transfer simulation.

Once the convective heat transfer from the working fluid to the pipe has been determined, the next step is to calculate the thermal gradient in the pipe wall. This is calculated by using finite elements within the thickness of the pipe wall. The equation for the temperature at a given radius inside a wall based on the heat flux into the wall is found by rearranging Equation (4) into Equation (5b) and substituting it into the Equation (5a) for temperature at a given radius as per Bergman et al. [12], resulting in the final form of the equation given by Equation (5) :

$$T_r = \frac{(T_1 - T_2)}{\ln(\frac{r_1}{r_2})} \ln(\frac{r}{r_2}) \; + \; T_2 \tag{5a}$$

$$T_2 = T_1 - (\frac{q \ln(\frac{r_2}{r_1})}{2\pi L k}) \tag{5b}$$

$$T_r = T_1 + \left(\frac{q}{2\pi L k}\right) ln \left(\frac{r_2}{r_1}\right) \left(\frac{\ln\left(\frac{r}{r_2}\right)}{\ln\left(\frac{r_1}{r_2}\right)} - 1\right) \tag{5}$$

For the above equations: T_r is the temperature at radius r, T_1 is the inner surface temperature, r_1 is the inner radius, and r_2 is the outer radius.

Equation (4) and Equation (5) both show that this model is not dependent on temperatures external to the pipe. The model assumes a one-dimensional radial heat transfer, which means the heat flowing out of the pipe wall to the exterior is equal to the heat entering the pipe wall from the working fluid. The independence from external conditions allow this model to be applied to any pipe that experiences thermal cycling.

As the model progresses, internal elements fail during each cycle and can no longer be considered as part of the structure, leading to a smaller area upon which stress can be applied. Prior to determining the temperature gradient in the cylinder wall, the model sets all broken elements to zero and assigns the inlet temperature to the first non-broken element on the interior. In the case of a greater external temperature, the code would be modified to search from the exterior rather than the interior.

Thermal and Pressure Stresses. Once the temperature for each element has been determined, the thermal stresses can be computed. Thermal stresses in the wall are expected to be higher on the side closer to the higher temperature, as the elements will see a greater difference in temperature during a cycle. Per Hetnarski [1], the radial, tangential (hoop), and axial stresses due to thermal factors are calculated by Equations (6), (7) and (8) respectively:

$$\sigma_{r,T} = \frac{E\alpha}{1-\nu}[\frac{1}{b^2 - a^2}(1 - \frac{a^2}{r^2}) \left(\int_a^b Trdr - \frac{1}{r^2}\int_a^r Trdr\right)] \tag{6}$$

$$\sigma_{\phi,T} = \frac{E\alpha}{1-\nu}[\frac{1}{b^2 - a^2}(1 + \frac{a^2}{r^2}) \left(\int_a^b Trdr + \frac{1}{r^2}\int_a^r Trdr - T\right)] \tag{7}$$

$$\sigma_{z,T} = \frac{E\alpha}{1-\nu}\left(\frac{2\nu}{b^2-a^2}\int_a^b Trdr - T\right) \tag{8}$$

For the above equations, α is the thermal expansion coefficient for the material, ν is Poisson's ratio for the material, a is the inner radius of the cylinder, b is the outer radius of the cylinder, T is the temperature at radius r, and E is the modulus of elasticity for the material. These equations operate under plane strain conditions, as the equations are designed for a thin-walled hollow cylinder with constrained ends that allow for zero strain in the z-direction. The temperature is assumed to be steady-state and does not reflect dynamic temperature variations, which is suitable for the long-cycle and time-independent approach of the ML-TFLAT.

These equations are valid for fixed-end scenarios, and may require modification depending on the case being modeled. Other cases for the equations can be found in Hetnarski [1]. These equations are dependent on both Eq. 5 and its integral across the width of the cylinder wall.

As pipe flow typically includes a pressure imparted on the cylinder walls, pressure terms were added and accounted for through the summation of principal stresses. The model assumes there will be nothing stopping the flow on either end of the modeled section, so the axial stress due to pressure is zero. When thin-wall conditions are found, radial and hoop stresses due to pressure are given by Equations (9) and (10) respectively as per Budynas et al. [4] :

$$\sigma_{r,p} = \frac{pD}{4t} \tag{9}$$

$$\sigma_{\phi,p} = \frac{pD}{t} \tag{10}$$

For the above equations: p is the pressure in the cylinder and D is the inner diameter of the cylinder, and t is the thickness of the cylinder wall.

For analysis in the model, the thermal and pressure stresses are summed and used to calculate a von Mises equivalent stress. Pressure is included in this analysis as the numerical model allows for the variation of pressure in the pipe. Axial, radial and hoop stresses are expected to be the primary stresses.

Accumulation of Damage. Once stress values for the cycle have been calculated for each element in the model, the damage caused by that thermal cycle can be calculated. Per Hussain [2], the maximum life of aluminum can be found using Equations (11) and (12) :

$$c = -0.0003T + 0.0805 \tag{11}$$

$$N_f = \left(\frac{\sigma_a}{651.8T^c}\right)^{-\frac{1}{0.092}} \tag{12}$$

For the equations above, c is a material-dependent coefficient that is experimentally determined, σ_a is the alternating stress, and T is the maximum operational temperature.

It is to be noted that the experiment described in Hussain [2] is not a one-for-one match to the use case being applied here. However, there is a lack of information on the thermal fatigue performance of 6061-T6 aluminum in the case described here. The constants in Equations (11) and (12) are material and environmentally dependent, but the general form of the equation is similar to Equation (6-16) in Budynas [4] , so the true relationship would likely be similar and this data can be used as an adequate substitute until further work is completed on this subject as this portion of the ML-TFLAT is being used solely for the purpose of this work to produce data for training the neural network component of the tool. Given that the response would be a similar equation, the patterns should still be representative of the real world and thus should still be able to prove the potential efficacy of the ML-TFLAT at its designed purpose.

Each element then increases the tracked damage by the inverse of the calculated lifetime at that temperature and stress per Miner's Rule, which states that the amount of the life of a material under variable stress levels that is used up by a cycle at a stress level is proportional to the number of cycles applied if that was the only stress the material experienced.

Failure Assessment. With damage accounted for, failure assessment takes place on both the nodal scale and on the heat exchanger tube as a whole. Each element is checked to determine if the damage accumulated has resulted in complete exhaustion of the fatigue life of that element. Once all elements have been checked, the model moves on to examine the whole component. The broken elements are used to determine the crack length that has formed, and this crack length is used to determine the stress intensity factor (K) by using Equation (13) as per Budynas et al. [4] :

$$K = \sigma_a \sqrt{\pi a} \tag{13}$$

For the equation above: σ_a is the alternating stress and a is the crack length.

The SIF is then compared to the fracture toughness of the cylinder material. If the SIF exceeds the fracture toughness, the material is considered failed.

In some situations, the SIF does not exceed the fracture toughness of the material before all elements have failed. In this case, the model additionally checks to see if all the elements have failed in the model and reports a complete failure as if the SIF had surpassed the fracture toughness.

Configuration Iteration. To analyse multiple configurations of a system, the model is designed to take inputs from a .csv file, which feeds the design information to the model iteratively. This allows a comparison to different operating conditions that can be used to determine expected longevities and validate results.

2.2. Data Generation and Preprocessing

As the TFM iterates through the lifetime of the model pipe, several data features are extracted and saved periodically to sequential .csv files for use both in model validation and

in training and evaluation of the LSTM network. This data must be processed prior to use, as the model produces vast amounts of data for each configuration that must be consolidated. Further preprocessing is needed to better suit the data to use within the LSTM network as the scale of numbers used has an impact on the stability of the algorithm. The process used is the same for both training and evaluation data as both must be in the same format for proper execution of the LSTM network.

2.2.1. Numerical Model Data Output

The four primary data outputs of the TFM are damage, strain, von Mises stress, and temperature of each element in the model. As the number of elements is increased, these files increase in kind. For the setup described above in this experiment, hundreds of thousands to millions of cycles were the typical range, leading to large file sizes. This, coupled with the necessity to back up modeled data as the model progresses, led to the implementation of a periodic export of data, selected to be at every 10,000 cycles. This number reduced the overall quantity of files and provided a frequent backup as well as the benefit to free up memory space. This allowed the model to run faster than previous attempts without the periodic saving mechanism in place.

Once the data had been produced by the TFM, a Python script is used to concatenate the files sequentially and to trim any trailing zeroes from the file for clean data usage. This data was then saved as a summary file, where it was used to produce the graphics for validation of the model.

The model configuration and options used for the TFM are saved with the generated data for future reference and use with the preprocessing of the training and evaluation data.

2.2.2. Data Used for Training the MLA

As the inputs used in this experiment do not include all the data produced from the TFM, new files for the training and evaluation data are generated using the strain and cycles from the TFM with the input parameters that are defined by the system. The cycles for the whole dataset are subtracted from the maximum cycle count for the model to get the remaining cycles information that will be used as the output for the training and evaluation process. This information is consolidated into a row for every cycle, which when combined with all other cycles, creates the data set to be used for training the MLA or evaluation of the MLA's performance.

2.2.3. Preprocessing of the Training and Evaluation Data

Further data processing is needed at this point to prepare the data for the LSTM network. Typically, these functions operate better when data values are within smaller ranges. For this, the cycles remaining are passed through a min-max scaler which converts the range of integer cycles remaining to a decimal between 0 and 1 representing the percentage of life remaining.

As the LSTM network operates on time series datasets, the data then must be collected into time series, where each sample in the series is one cycle further into the failure data than the previous one. The length of these time series samples is determined by the sample

rate, and the samples are grouped into batches. The size of the batches is determined by the number of batches specified in the import options, but it is preferable for LSTM stability to keep the batch count between 20 and 30 based on testing of different batch sizes. [22] For this experiment, a batch size of 25 was found to be effective at getting consistent training. The generation of the time series is handled through a Keras preprocessing function. With the data sets processed, they are suitable for use with the LSTM network.

2.3. Machine Learning Algorithm Design Process

2.3.1. Selection of Candidate Algorithms

With the implementation of a machine learning algorithm (MLA), there are several factors to consider as to how the MLA will function. Beyond the general type of MLA, there are several hyperparameters to consider as well. As discussed by Ayodele [3], several options that were considered are given below.

Algorithm Type. For this experiment, several different types of MLAs were considered as there are benefits and drawbacks to each available option.

Linear Classifiers are a staple of classification problems that strive for grouping data accurately into specific categories. Typically, this is done through the implementation of a "hyperplane" that bisects a data space to divide points into their respective correct regions. For this experiment though, classification is not the preferred method as the goal of this tool is to provide a predicted remaining life, not to categorize the data into one of many sets.

K-means clustering is another type of machine learning algorithm that performs well in unsupervised learning scenarios through a simple means of implementation and a mechanism of grouping data based on clustering closely related data. This algorithm type however functions as a classification type algorithm as well and is used for unsupervised learning, while this experiment is designed to utilize supervised learning.

Neural networks are highly versatile MLAs that can handle complex and non-linear relationships through multi-layered and varied architectures. They are capable of both categorization and regression, the latter of which is needed for this experiment. The drawback for neural networks, however, is the amount of training data needed, the time it takes to train them, and the myriad of hyperparameters that must be adjusted without clear guidance. Despite these drawbacks, a neural network was deemed to be the best fit for this application. Furthermore, neural networks are well-documented and considerably common, so the implementation process is easier and supported by knowledge and experience widely available on the internet.

Neural Network Type. With a neural network selected as the MLA for this experiment, the type of neural network was the next decision made. The type determines how the neural network functions and therefore it is important to select an appropriate type.

The simplest form of a neural network is the deep feed forward network. This type of network utilizes basic cells that feed data forward through multiple layers. Adding layers can increase the capacity for detecting complicated patterns, but also exponentially increases the time required to train. While this option potentially could be successful in this research, there were other options to examine as well.

More specialized structures such as autoencoders and generative adversarial networks exist, but tend to serve a special purpose, such as providing the capacity for unsupervised learning or generation of data. The recurrent network was the network type selected for this research. A recurrent network implements layers of special cells that have the capacity to recall data from previous inputs and use that as another input for making decisions regarding future inputs. More specifically, an LSTM network uses special cells called long-short term memory cells (LSTM) that are better suited for handling larger gaps in changes in data. As high-cycle fatigue requires a large number of cycles and the typical three phases of crack development as described in Shigley's [4], large periods of minimally changed data are expected, especially during the early stages of fatigue onset. This suits the LSTM type network better than other alternatives and was selected for this research.

2.4. Selection of Hyperparameters

2.4.1. Architectures

The architecture, or arrangement of layers, of an LSTM network has significant implications on its ability to learn from datasets. There is, however, no one-size-fits-all method for determining the best architecture to use for a given dataset, so experimentation is needed. For this research, two types of layers were selected to be used in different configurations.

Dense (D) layers consist of cells that are fully connected to both the input layer of cells and the following layer of cells. These provide the basic building block for many neural networks.

LSTM (L) layers are more complicated in that they retain a memory of previous values that have passed through them. This memory serves as an additional input for the cell, allowing for data to be analyzed in time series rather than in snapshots. For a long, time-dependent process like fatigue, a time series analysis is needed to show that the amount of cycles that have passed has an influence on the future state of the system. For a LSTM network, at least one LSTM layer must be present.

Each layer added to the system increases the number of parameters that need to be trained, which in turn increases the training time. LSTM layers have significantly more parameters due to their more complicated nature, so limiting the number of LSTM layers can reduce the required training time for that system architecture. It is important to note also that the smallest network is not guaranteed to function the best, as shown by Greenwood [5]. Listed below in Table 2 are the architectures that this experiment tested.

Table 2: Examined LSTM Architectures

Architecture Name	**Layer Layout**	**Total Parameters**
Arch-1	D-L	58
Arch-2	L-D	226
Arch-3	L-L	248
Arch-4	D-L-L	278

2.4.2. Activation Function

The function that determines the response of each cell within a neural network, called the activation function, is one of the easier hyperparameters that can be used to alter performance of the network. As several functions exist for this purpose and the implementation is inexpensive, two activation functions were tested in this research to observe the effect on the LSTM network performance, shown in Table 3.

Table 3: Examined Activation Functions

Activation Function	**Output Range**	**Features**
Sigmoid	(0,1)	Differentiable, monotonic
Tanh	(-1,1)	Differentiable, monotonic

2.4.3. Initializers, Loss Function and Optimizer

Further optimization of the LSTM networks can be sought using different initializers for the layers, different loss functions for the metrics of the training, and different optimizers that adjust the training mechanism. These methods have lesser impacts on the function of the LSTM network and modifying more variables increases the complexity of the analysis. Therefore, the Adam optimizer, mean squared error, and a 'ones' initializer were used in this research as they provided more consistent results. They were also recommended in documentation for regression networks.

2.5. Implementation of Selected Algorithm

Neural networks can be built from scratch, but this would require extensive calculus and large scripts that are prone to errors when written without rigorous and extensive quality assurance efforts. This is not a beneficial task for the scope of this work, so alternatives were researched. The programming language Python was selected for its affinity to data science, the understandability inherent in the code structure, and the plethora of libraries available for assisting this implementation including a specific library called Keras [6].

Keras is a high-level Python library for implementing neural networks with the capacity to utilize multiple different backends to process the model without requiring a deep understanding of the mechanics of neural network programming. The default backend, Tensorflow, was selected as well for its natural integration with Keras and its effectiveness at utilizing GPU acceleration during training. Tensorflow performs the required calculations needed to use neural networks through the use of tensors.

The program was developed in an object-oriented manner to facilitate building, training, and evaluating multiple models per run which, in turn, increases the time efficiency of the program. By allowing the program to run day and night, more models were able to be produced for the analysis portion of this experiment. To help facilitate this automation, instructions are pre-set in .csv files that identify the options to be used during all stages of the development of the LSTM models as well as a file that dictates the architecture of each

model. These separate files are intended to provide isolated control of the function of the program to reduce the chance of user error that could render the program inoperable.

Outputs from this implementation include the training history of the loss function output and accuracy metric of the model, the random seed used to set the random number generator that was used to initialize the starting weights for repeatability, the date of training, the total elapsed time of training and the number of training iterations in which the loss function did not improve. Evaluation data outputs include a list of descriptive statistics for both the loss and accuracy metrics and the loss and accuracy history for each evaluation dataset.

2.6. Supervised Training of the LSTM Network

Training of a neural network can be supervised or unsupervised. Unsupervised training involves using input data with no correlated output data. This can be useful for systems with unknown outcomes, but it is not particularly useful for this application. The LSTM networks analyzed in this research were all trained using supervised training techniques, where the output of the remaining cycles from the TFM was used as a metric for the LSTM to measure its predictions against.

Six datasets were used for the training of these networks, each adjusting the value of one variable at a time to avoid confounding the effect of one variable on another. The variables examined are the designated input variables without the inclusion of the strain, as that is the primary indicator for how the other variables affect the system. By training each LSTM network using this collection of training data, we can tune the parameters based on each of the input values.

To assist with the training process this research used the following methods: a checkpoint saving method, an early stopping method, and a learning rate reduction method. The checkpoint saving method saves the best version of the model as the system trains, ensuring that the best model is retained and safeguarding against losing data from an unexpected program failure such as power outages or system crashes. The early stopping method stops training on a dataset when training has stagnated for a given number of epochs, or training iterations over the training data. For this research, the number of epochs to wait was set to 10. The learning rate reduction method reduces the learning rate when the training stagnates and is a recommended practice to enhance the learning process. For this research, the learning rate was reduced by 50% when five epochs pass with no improvement to the loss metric. This allows the learning rate to be adjusted in the middle of the early stopping wait period to allow for any benefits of the reduction of learning rate to take effect before the training process moves to the next data set.

The training process for an LSTM network requires many calculations that scale up rapidly with increased sizes of the network. Therefore, it has become common practice within the Machine Learning industry to implement graphics processing unit (GPU) acceleration for training. This technique was implemented for this research, resulting in a ten-fold reduction in training time using a Nvidia Geforce 1070 GPU.

2.7. Evaluating the LSTM Network

As the goal of this research is to determine the validity of using an LSTM network in combination with the TFM, the LSTM network must be evaluated using metrics. The primary metrics needed are included with the training process of an LSTM: the mean squared error of the network and the accuracy of the network. Furthermore, Keras includes a method for evaluating neural networks on input data which is provided by the TFM.

By running the evaluation method with each evaluation data set on each model architecture and configuration, we are able to get a loss value and accuracy value for each set that serves as an evaluation metric for the performance on that data set. These metrics will be aggregated and used in the analyses in the Results section.

3. Results

The metrics of the results of this research include components for both the TFM and the LSTM networks evaluated. For the ML-TFLAT to be considered a potentially viable tool for the prediction of thermal fatigue life, two requirements must be fulfilled:

1. The TFM must be shown to be accurate and valid for the system that it is modeling.
2. The LSTM Network must show that there exists at least one network configuration that provides meaningful data.

The TFM is evaluated to determine the accuracy of the model at predicting thermal fatigue, and the LSTM networks are evaluated on their loss metrics and reported accuracy from the evaluation data.

3.1. Accuracy of the TFM

As the TFM is a finite element model approximating a real-world system, the model has several accuracy requirements that must be met for the TFM to be considered a valid tool:

1. The TFM must be verified as a solution that increases in accuracy asymptotically with a greater number of elements.
2. The TFM must accurately represent the expected real-world scenario, modeling the structure, the material properties, the fatigue properties, and the thermo-mechanical response of multiple system configurations.

To evaluate the numerical validity of the model, the convergence of solutions and the effect of failed elements will be examined. For the real-world representation evaluation, the effect of failed elements will again be used in comparison to the analysis of how the system responds. The TFM response will be compared to literature and examining if modifying the parameters of the model corresponds to the expected change in output that would be seen in a real-world system.

3.1.1. Model Convergence

For a finite element model to be considered accurate, the solution must trend towards a single solution as the number of elements increases. This represents the transition from a discrete model to a continuous model. For this evaluation, the number of cycles until failure

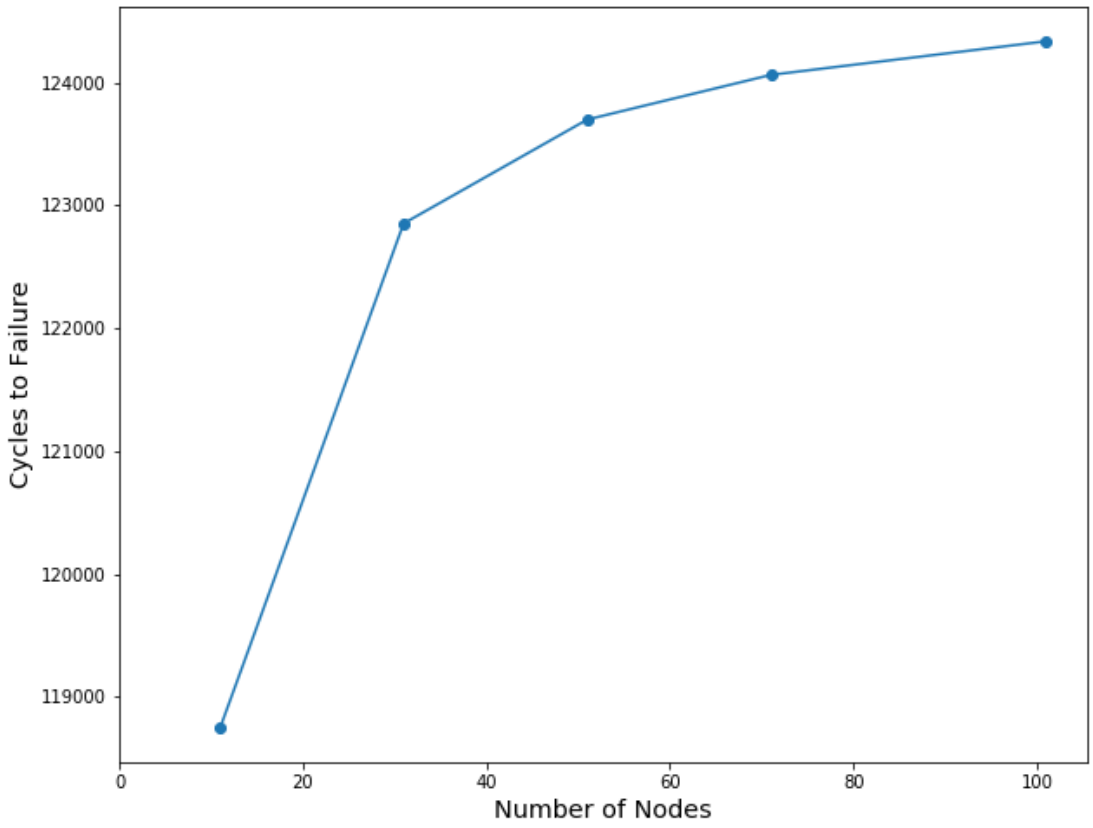

Figure 1: Model Convergence as number of nodes is increased

was found by running the TFM with increasing numbers of elements, followed by plotting the number of elements against the cycles to failure as seen in Figure 1.

The data in Figure 1 indicates that as the number of elements increases, the cycles to failure increases as well, but in a decreasing amount per added element. This shows that the model is converging towards a solution as the number of elements increases. Additionally, this data provided insight that assisted in selecting the element count of 61 that was used in the experiment, as it visualizes the tradeoff between accuracy of model data to the model run time.

3.1.2. Effect of Failed Elements

As the TFM progresses through the cycles in its calculations, some elements begin to reach their maximum allowable damage and therefore are marked as "failed", indicating the material at that element no longer can bear stress and also that a crack has started to form at that point. With each failed element, the stress load on the other elements is expected to increase. Given that the greatest temperature gradient per cycle is located at the interior surface of the pipe wall, the crack is expected to initiate internally and then propagate radially outward.

To check for this phenomenon, a damage plot with information from elements throughout

the pipe wall can be constructed, seen in Figure 2:

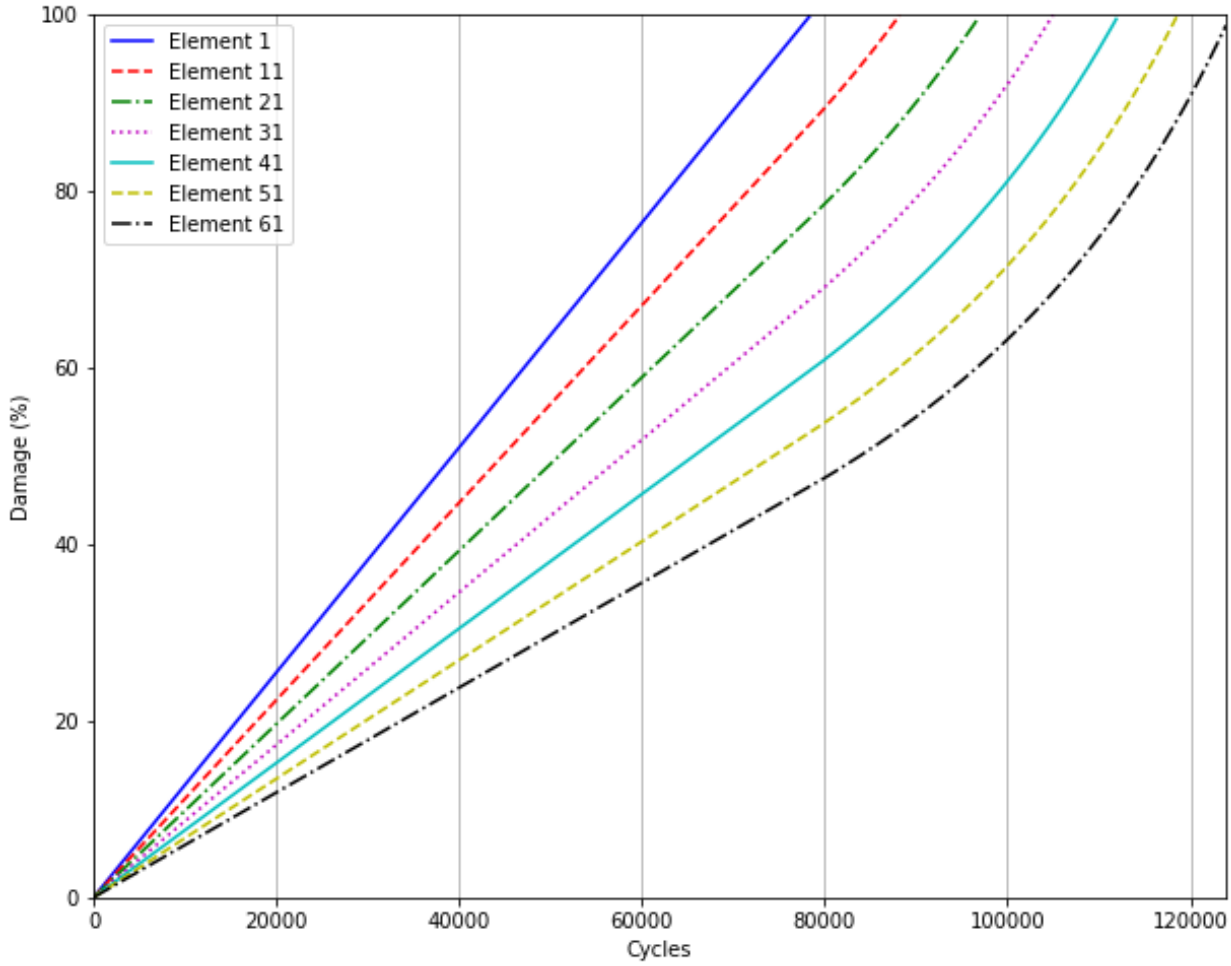

Figure 2: Damage of Elements Through the Wall Thickness at Selected Element Locations

This figure shows the cumulative damage at each cycle for 7 elements, starting with Element 1 at the surface of the inside wall and ending with Element 61 on the exterior surface. As is expected in the real-world case, Element 1 fails first, leading to the failure of each element sequentially moving outward towards Element 61. Additionally, the expectation that the surviving elements experience greater stress loads as internal material fails is shown by the increase in rate of damage experienced as earlier elements fail. Element 61 exemplifies this particularly well, as it is subjected to the most impact over the course of the model's runtime.

3.1.3. Comparison to Literature

Fatigue behavior can be described, per section 6-6 of Shigley's [3], as taking place over three phases: the plastic deformation phase (Phase 1), the crack propagation phase (Phase 2), and the unstable crack phase (Phase 3).

For the TFM to act as an appropriate model for this system, the stress of Element 61 from the baseline configuration used in the Parameter Variation Analysis can be examined to detect characteristic changes that correspond to the transitions between phases. Throughout Phase 1, the stress load is not changing for the element and should remain constant. Phase 2 can be denoted by the initiation of deviation from the constant stress load, and Phase 3 is characterized by distinctly greater rates of change of stress load than at the beginning of Phase 2, as shown in Figure 3.

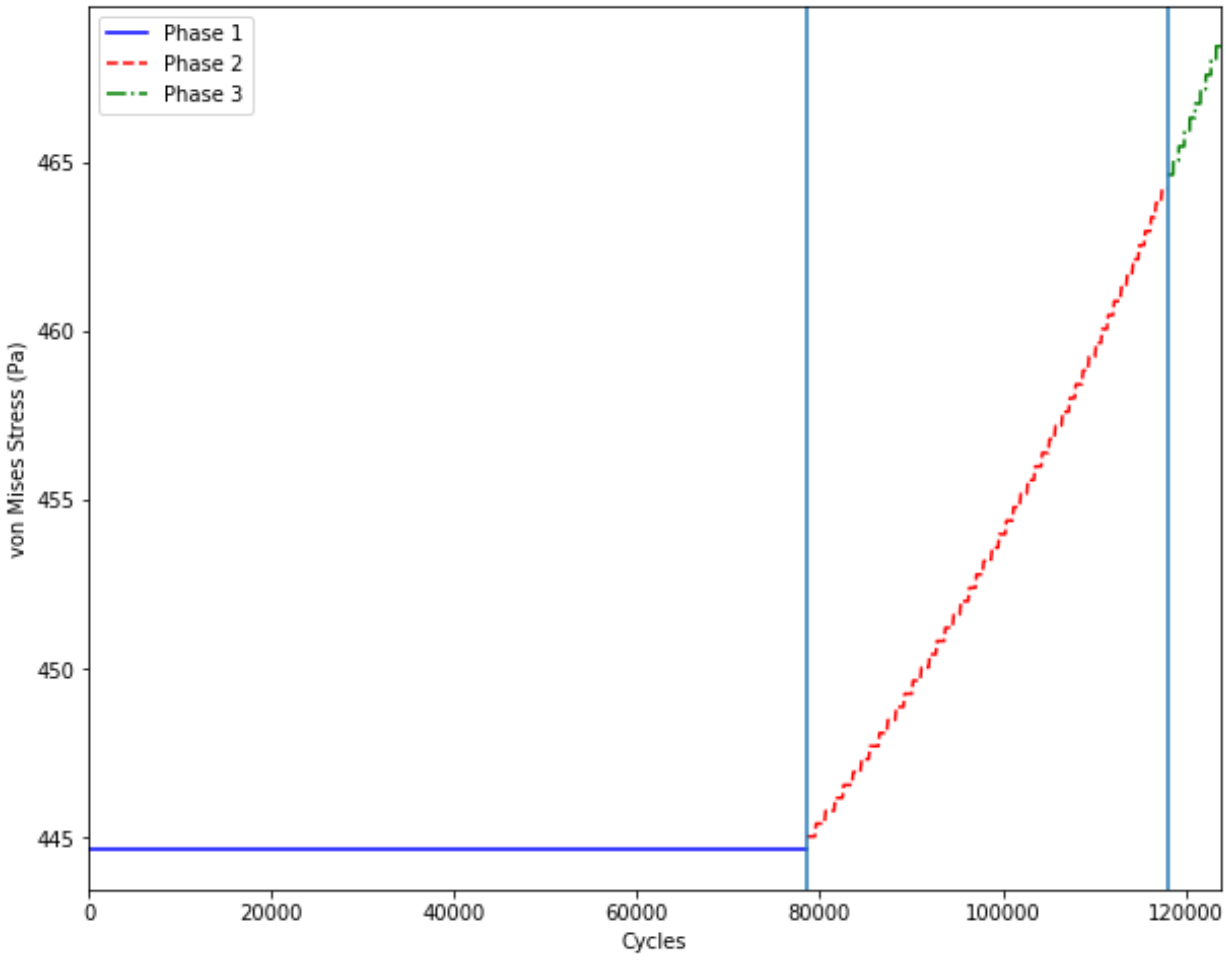

Figure 3: von Mises Stress in Element 61 – Baseline Case

The expected changes can be seen in Figure 3, indicating a shift to Phase 2 near 78,000 cycles and the shift to Phase 3 occurring near 115,000 cycles.

3.1.4. Effects of Parameter Variation

The TFM is expected to operate on a range of system configurations. The effect of changing input parameters to the model must be analyzed to confirm that the model responds in the manner expected according to the way a real-world system would react. For

each parameter, three different system configurations were tested, with a common baseline configuration used in all data sets. Variables were changed independently, and the collective set of values used are shown in Table 4.

Table 4: Parameter Values for Variation Analysis

	Baseline	Var-1	Var-2
Inlet Temperature (°C)	95	92.5	90
Outlet Temperature (°C)	75	70	65
Wall Thickness (m)	0.0017	0.0033	0.0049
Flow Pressure (Pa)	0.015	0.15	1.5
Flow Rate (m/s)	0.001	0.003	0.005

Pipe Inlet Temperature. For a heat exchanger system, an increase in the pipe inlet temperature is expected to increase the temperature at the internal surface of the pipe. With a higher internal surface temperature for each cycle, the thermal stress increases, which increases the strain and fatigue damage rates. The results of the analysis are shown in Figure 4.

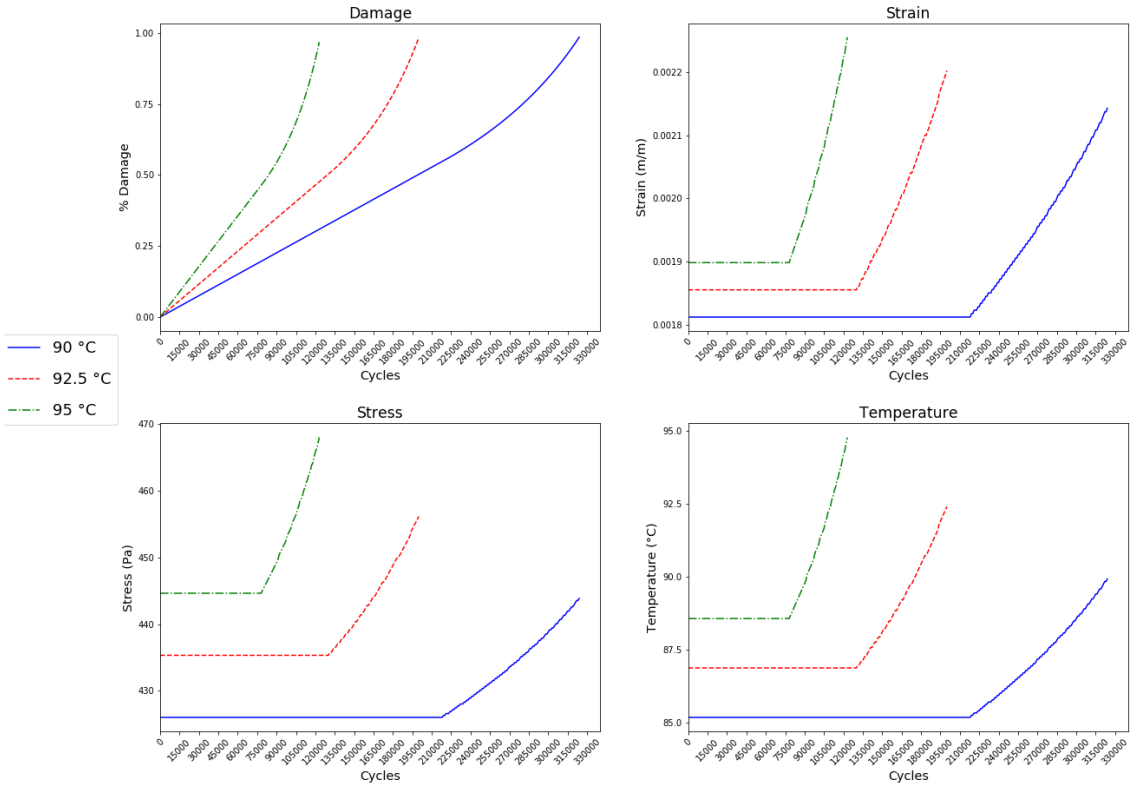

Figure 4: Effects of Varying Inlet Temperature on Thermal Fatigue Life

The data shows that the thermal life is reduced as inlet temperature increases, along with the expected increase in the stress and strain, and the expected decrease in Element 61 temperature. The temperature plot shows that Element 61 reaches higher temperatures quicker than the baseline, which provides evidence as to why the thermal fatigue life increases. A higher temperature leads to greater stress ranges from thermal expansion. The data retained the expected shape from the literature comparison.

Pipe Outlet Temperature. For a reduction in outlet temperature with all other examined variables held constant, the heat exchanger is expected to have a greater rate of heat transfer out of the external pipe surface. With a higher heat transfer rate, the temperature at the surface of the material is expected to be lower and the thermal gradient through the pipe wall is bigger. This leads to smaller stresses and strains in elements further from the internal surface where the temperature, thermal stress, and strain are smaller. Reducing that stress on an element closer to the surface allows it to accumulate less damage prior to the crack reaching that element. The results of this analysis are shown in Figure 5.

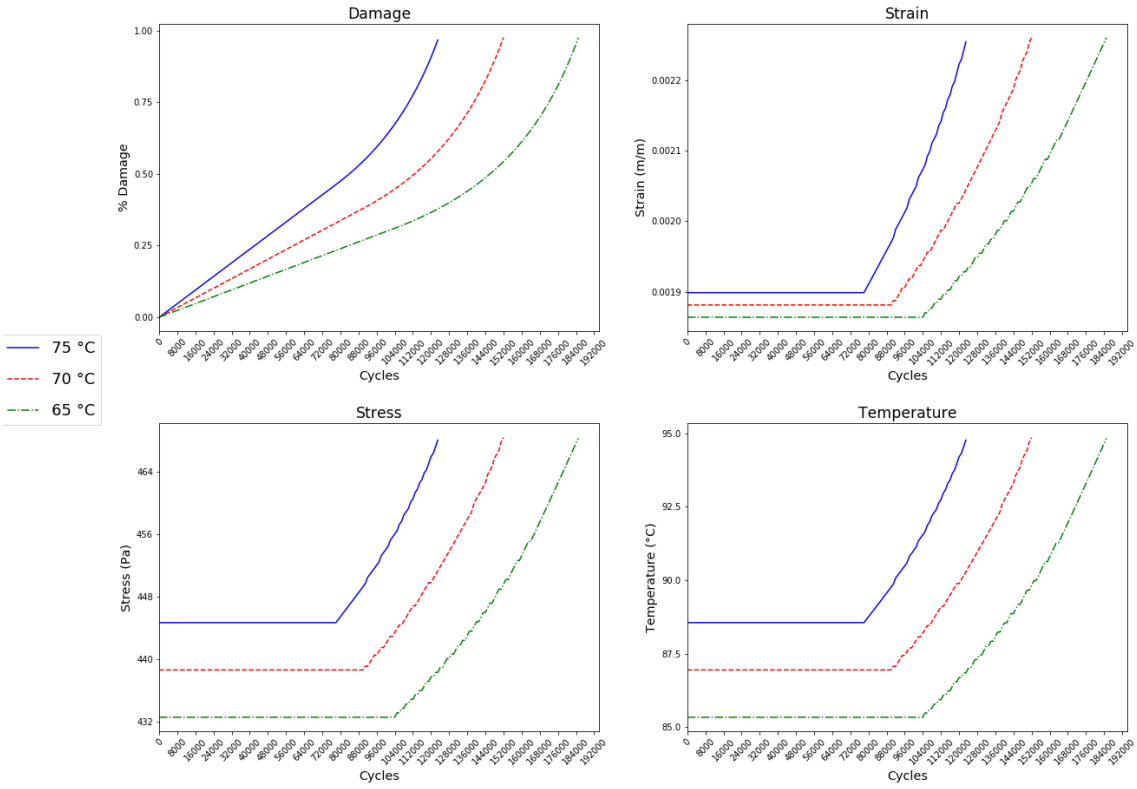

Figure 5: Effects of Varying Outlet Temperature on Thermal Fatigue Life

These results indicate that the expected behavior was found in the model, with longer thermal fatigue lifespans, lower stress and strain values, and lower Element 61 temperatures. The more exaggerated upturn in damage as the crack grows shows that the elements are not subject to as much damage during Phase 1 than in other cases. The data retains expected

literature shapes, indicating no abnormalities with how the model is numerically representing the data.

Pipe Wall Thickness. To accommodate the model's inputs, varying the wall thickness was completed by increasing the outer radius from 0.0127m to 0.0159m while maintaining the inner radius at 0.011m. Increasing the wall thickness of the pipe introduces more material for more distributed handling of stresses and a larger thermal resistance to increase the thermal gradient across the pipe wall, effectively reducing the element temperatures as they are further removed from the interior surface. The lower stresses due to the higher load area reduces strain as well. The increased thickness also increases the chance of a critical crack forming and propagating through the material rapidly. The results of this analysis are shown in Figure 6.

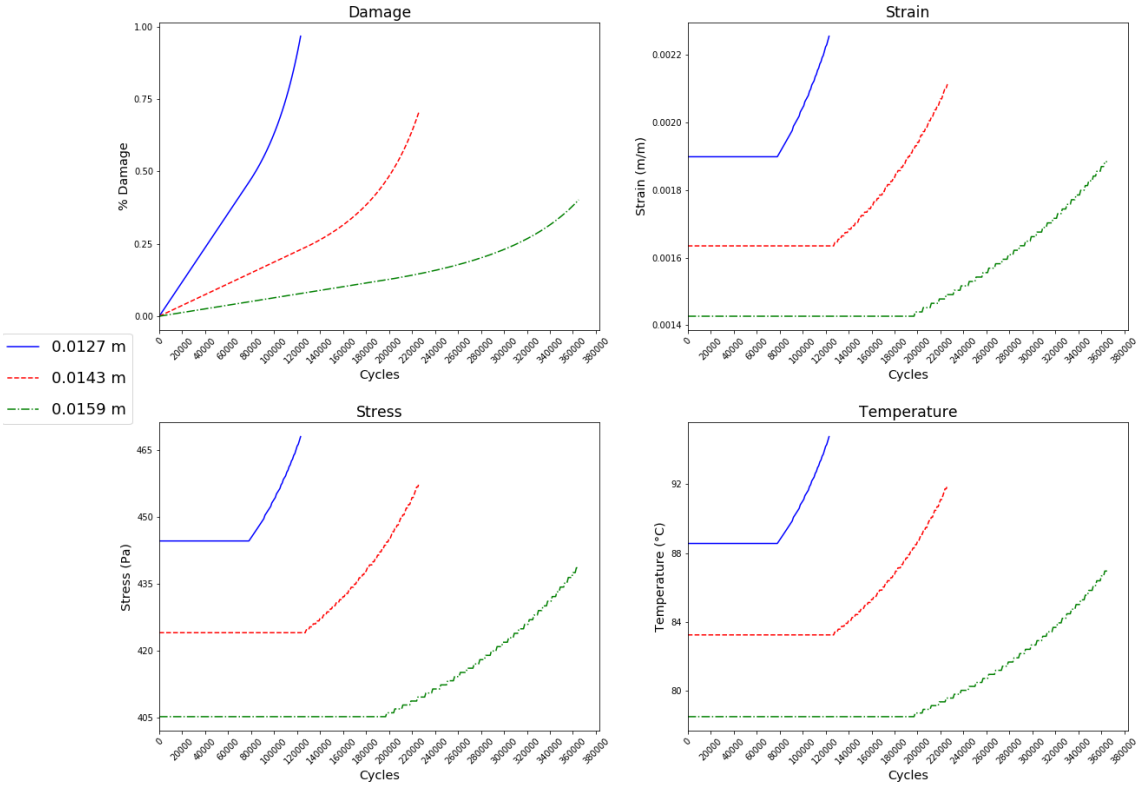

Figure 6: Effects of Varying Outer Diameter on Thermal Fatigue Life

Compared to the other figures from this analysis, Figure 6 shows that the model represents critical cracks well through the failure of the material prior to the 100% damage point on Element 61. The resultant stress, strain, and exterior surface temperature decrease with an increased wall thickness as expected, and the data retains the expected shape in all cases.

Fluid Flow Pressure. Flow pressure does not contribute to the thermal loading of the pipe, but it does impact the overall stress state. Increasing the pressure applies radial and hoop

stresses that combine with the thermal stresses to lower the von Mises stress by pushing the stress state closer to a hydrostatic state. This reduces the damage accumulation, lowers the stress and strain of the system, and delays the onset of temperature shifts. The results of this analysis are shown in Figure 7.

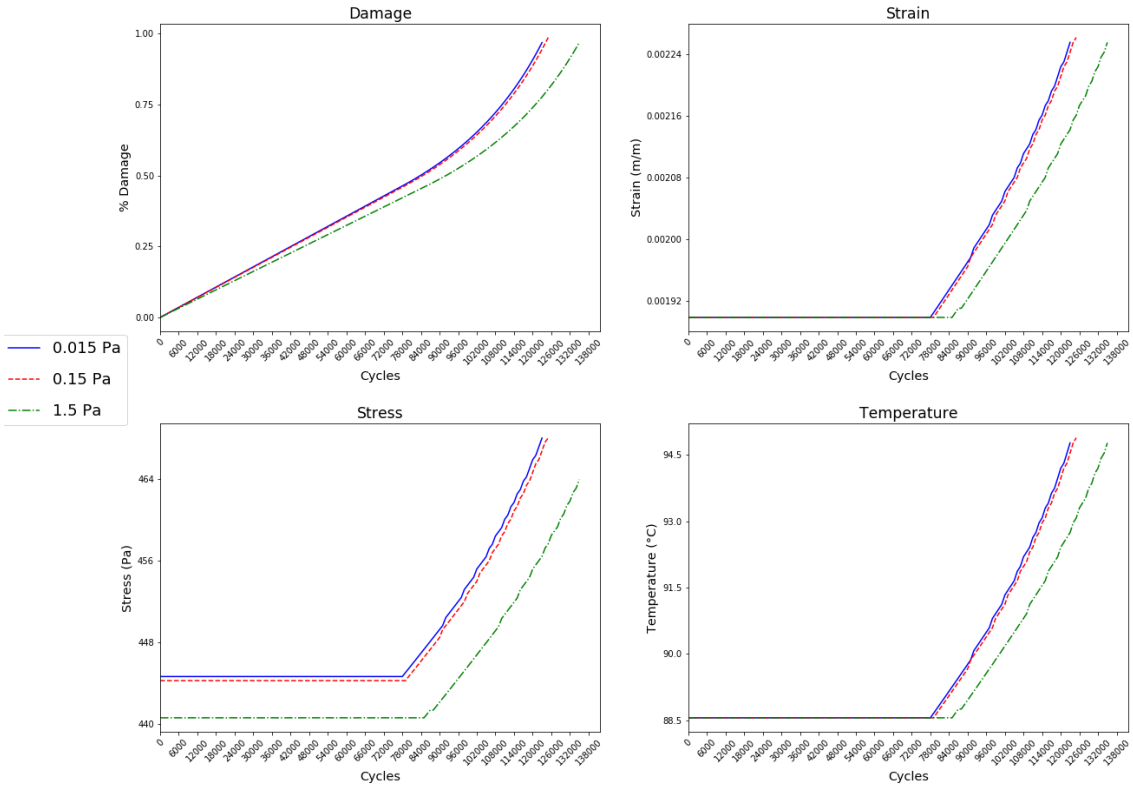

Figure 7: Effects of Varying Fluid Flow Pressure on Thermal Fatigue Life

The results of varying pressure shown in Figure 7 were as expected. There was no change in the thermal component indicated by the temperature other than the delay caused by the failure of elements from fatigue. The von Mises stress relationship is nonlinear, which is represented by the nonlinear rate of change between test cases. The data shows the expected phase response.

Fluid Flow Rate. For convective heat transfer that is found inside the pipe, increasing the flow rate is expected to increase the heat transfer into the pipe wall from the fluid. This indicates that the expected results should be similar to the Outlet Temperature results, with decreased damage accumulation rates, stress, strain and external surface temperatures as flow rate increases due to faster heat rejection from the pipe wall. The results of this analysis are shown in Figure 8.

As expected, the results in Figure 8 respond as expected. Of note is the exaggerated Phase 2 and Phase 3 transitions in the 0.005 m/s case. This clearly shows the effect of a

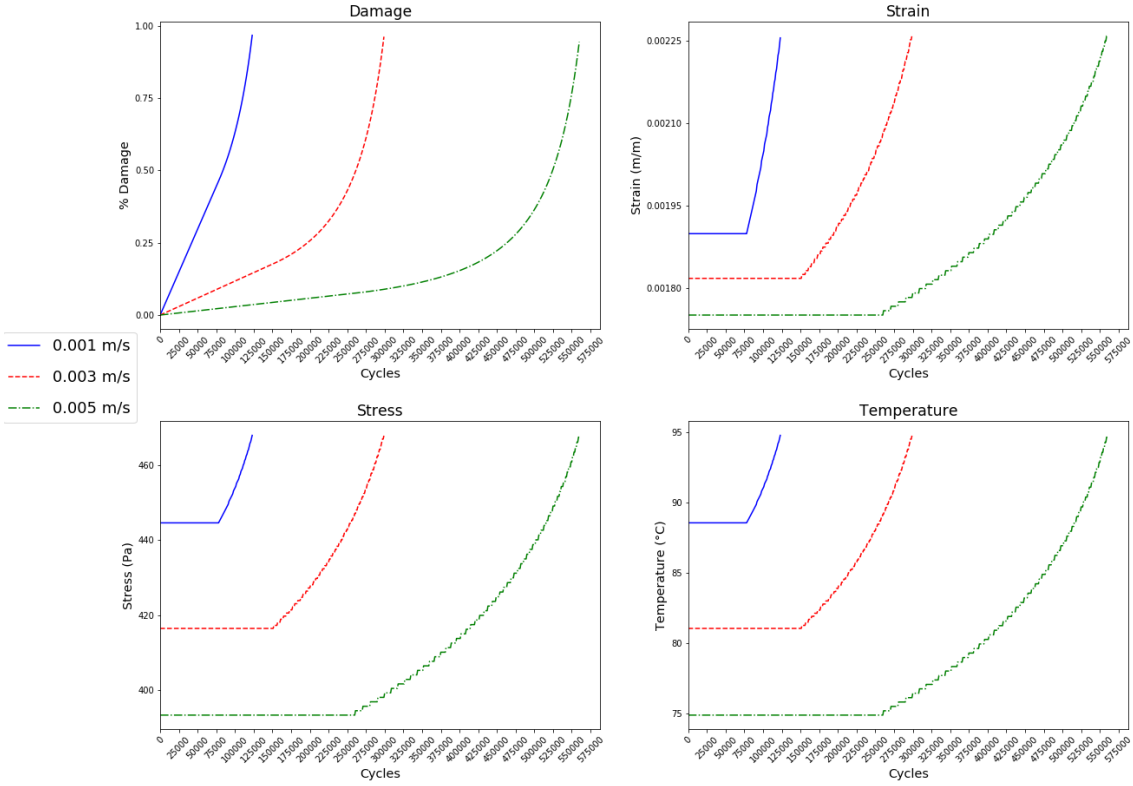

Figure 8: Effect of Fluid Flow Rate on Thermal Fatigue Life

bigger gradient through the wall as the rate of damage accumulation is significantly slower than the baseline, increasing the time until destruction.

3.1.5. Summary of TFM Results

The model convergence analysis shows that the TFM does represent a valid numerical solution by proving the solutions converge towards what would be the continuous system solution. The results of the failed element analysis show that the TFM accurately represents the equations governing the elements. These two results provide clear evidence of the numerical accuracy of the TFM and fulfills the first accuracy requirement for the TFM.

The failed element analysis shows that the TFM accurately represents the equations of interaction between nodes in the material, therefore representing the model realistically on the nodal level. Comparison to information provided in Shigley's regarding the expected phases of fatigue indicates that the model responds to stress correctly when compared to aluminum fatigue information, indicating the accuracy of the TFM in terms of the material and fatigue property representation. Analyzing the effects of parameter variation shows that the TFM responds as expected to different system configurations, thus providing proof of the robustness of the thermal model as well as the TFM's capacity to accurately represent a range of system configurations as needed for the LSTM Network portion of the ML-TFLAT. These results combined prove the real-world representation accuracy of the TFM, which

fulfills the second requirement of the TFM.

3.2. Viability of the LSTM Networks

For the viability of the LSTM network as a potential tool to be proven, two requirements must be fulfilled:

1. At least one network architecture must show accuracy for at least part of the evaluation data.

2. Any network that shows evaluation data accuracy must not be subject to overfitting.

These requirements will be assessed using performance data from the evaluations and statistical analysis.

3.2.1. Model Performances

The mean squared error loss and accuracy data for each of the 40 evaluation data sets was collected for each of the LSTM architectures and configurations. A summary of this data is included in Table 5.

Table 5: Summary of LSTM Evaluation Data

	Loss		**Accuracy**	
	Mean	**Variance**	**Mean**	**Variance**
Tanh				
Arch-1	0.720672	0.001455	0	0
Arch-2	0.369411	0.000942	0.175	0.148077
Arch-3	3.84E-06	0.008447	0.3	0.215385
Arch-4	0.34363	1.36E-18	1	0
Sigmoid				
Arch-1	0.58507	1.57E-07	0	0
Arch-2	0.381674	4.89E-07	0	0
Arch-3	0.345478	1.73E-16	0	0
Arch-4	0.524296	4.5E-09	0	0

The Sigmoid activation function in Table 5 failed to show any degree of accuracy with the evaluation data. However, the Tanh activation function was able to display accuracy in three of the four network architectures. Arch-2 and Arch-3 were successful on only a selection of the evaluation data sets, but Arch-4 showed accuracy on all sets. This result is unlikely for a stochastic system, and likely indicates a case of overfitting.

3.2.2. Model Comparisons

To fully evaluate neural networks, data must be collected on the performance of the network on multiple data sets to provide a statistically supported conclusion. As one statistic is insufficient to fully describe a neural network's performance, the mean and variance was calculated for each network's loss and accuracy results from each of the 40 evaluation data sets. Additionally, to compare the performance of each network against the others, ANOVA

was used to determine if the networks performed differently in a statistically significant way. These metrics were selected as suggested by Flexer. [10] The results of the ANOVA performed on the loss and accuracy data are shown in Table 6 and Table 7 respectively.

Table 6: Loss ANOVA Summary

ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Sample	7.138515	3	2.379505	1755.309	6E-195	2.633547
Columns	1.067203	1	1.067203	787.2523	2.53E-87	3.871436
Interaction	4.998765	3	1.666255	1229.16	2E-172	2.633547
Within	0.422949	312	0.001356			
Total	13.62743	319				

Table 7: Accuracy ANOVA Summary

ANOVA						
Source of Variation	SS	df	MS	F	P-value	F crit
Sample	11.53437	3	3.844792	84.6261	4.38E-40	2.633547
Columns	10.87812	1	10.87812	239.4339	1.77E-40	3.871436
Interaction	11.53438	3	3.844792	84.6261	4.38E-40	2.633547
Within	14.175	312	0.045433			
Total	48.12188	319				

With $\alpha = 0.05$, both Table 6 and Table 7 shows that all P-values are dramatically less than 0.05, indicating that the architectures are significantly different. This in conjunction with the performance data shows that there are potentially distinct architectures and configurations that can provide predictions on the remaining thermal fatigue life.

3.2.3. Summary of LSTM Results

The model performance analysis indicated that the tested architectures and configurations were able to produce some accurate predictions without overfitting. The model also provided examples of poor fits and an example of an overfit. This fulfills the second requirement of the LSTM network viability analysis.

The ANOVA results show that there are statistically independent distributions within the selected network architectures and configurations, indicating that distinct networks can be generated that can produce results with valid accuracies. This fulfills the first requirement of the LSTM network viability analysis.

4. Conclusions

Per the requirements of both the TFM results analysis and the LSTM results analysis, this research has proven two things. First, it proves that the TFM model produced is an accurate model that well represents the thermomechanical response that leads to thermal fatigue. Secondly, it proves that there exists at least one LSTM network that can provide predictions of thermal fatigue life without overfitting to the training data. With these two results in mind, the data presented here indicates that the ML-TFLAT concept is viable, indicating that further research into this tool could provide a novel method for improved thermal fatigue life analysis.

References

[1] Hetnarski RB, Eslami MR. Chapter 6: Disks, Cylinders and Spheres. In: and others, editor. Thermal Stresses – Advanced Theory and Application;. .

[2] Hussain F, Abdullah S, Nuawi MZ. Effect of temperature on fatigue life behaviour of aluminium alloy AA6061 using analytical approach. J Mech Eng Sci ISSN. 2016;10(3):2324–2235.

[3] Ayodele TO. Types of Machine Learning Algorithms. Intech Open; 2020. Available from: www.intechopen.com.

[4] Budynas RG, Nisbett JK. Shigley's Mechanical Engineering Design; 2011. .

[5] Greenwood M, Oxspring R. THE APPLICABILITY OF 'OCCAM'S RAZOR' TO NEURAL NETWORK ARCHITECTURE;.

[6] Chollet F, and O. Keras; 2015. Available from: https://keras.io.

[7] Li X, Ding Q, Sun JQ. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliab Eng Syst Saf. 2017;172:1–11.

[8] Schwartz MP. Four Types of Heat Exchanger Failures . . . mechanical, chemically induced corrosion, combination of mechanical and chemically induced corrosion, and scale, mud, and algae fouling; 1982.

[9] Subramanian RS. Heat transfer in Flow Through Conduits;.

[10] Flexer A. Statistical Evaluation of Neural Network Experiments: Minimum Requirements and Current Practice. Cybernetics and Systems Research. 1996;.

[11] Ahmadzadeh F, Lundberg J. Remaining useful life prediction of grinding mill liners using an artificial neural network. Minerals Engineering. 2013;53:1–8. Available from: https://dx.doi.org/10.1016/j.mineng.2013.05.026.

[12] Bergman TL, Lavine AS, Incropera FP, Dewitt DP, et al. Fundamentals of Heat and Mass Transfer, 7th Edition. 7th ed. 111 River Street, Hoboken, NJ 07030: John Wiley & Sons; 2011.

[13] Guan X, Guan X, Jha R, Jha R, Liu Y, Liu Y. Probabilistic fatigue damage prognosis using maximum entropy approach. Journal of intelligent manufacturing. 2012;23(2):163–171.

[14] Nakayama T, Yuse F, Tsubokawa Y. Direct observation of corrosion fatigue cracks in aluminum alloy using ultra-bright synchrotron radiation. Corrosion science. 2007;49(1):130–138.

[15] Wang XS, Xu Y, Xu XQ. Direct Observation of Fatigue Cracking in the Fuel Plate Using the Scanning Electron Microscope. Applied composite materials. 2004;11(3):145–154.

[16] Ravi P, Naragani D, Kenesei P, Park JS, Sangid MD. Direct observations and characterization of crack closure during microstructurally small fatigue crack growth via in-situ high-energy X-ray characterization. Acta materialia. 2021;205:116564.

[17] Santecchia E, Hamouda AMS, Musharavati F, Zalnezhad E, Cabibbo M, Mehtedi ME, et al. A Review on Fatigue Life Prediction Methods for Metals. Advances in Materials Science and Engineering. 2016 Sep;2016:9573524.

[18] Foundation NS. NDT Resource Center;. Available from: https://www.nde-ed.org/EducationResources/CommunityCollege/communitycollege.htm.

[19] Company TP. Sensors - A Complete Guide;. Available from: https://www.thomasnet.com/articles/instruments-controls/sensors/.
[20] Microelectronics ST. Industrial Sensors;. Available from: https://www.st.com/en/applications/factory-automation/industrial-sensors.html.
[21] Embitel. 7 Most Commonly Used Sensors for Developing Industrial IoT Solutions;. Available from: https://www.embitel.com/blog/embedded-blog/7-most-commonly-used-sensors-for-developing-industrial-iot-solutions.
[22] Masters D, Luschi C. Revisiting Small Batch Training for Deep Neural Networks; 2018.