

Electronic Thesis and Dissertation Repository

4-24-2023 10:30 AM

Data-Driven Predictive Maintenance: HVAC Health Prognostics Using Power Consumption and Weather Data

Ruiqi Tian, *Western University*

Supervisor: Capretz, Miriam, *The University of Western Ontario*

A thesis submitted in partial fulfillment of the requirements for the Master of Engineering Science degree in Electrical and Computer Engineering

© Ruiqi Tian 2023

Follow this and additional works at: <https://ir.lib.uwo.ca/etd>



Part of the [Artificial Intelligence and Robotics Commons](#), and the [Software Engineering Commons](#)

Recommended Citation

Tian, Ruiqi, "Data-Driven Predictive Maintenance: HVAC Health Prognostics Using Power Consumption and Weather Data" (2023). *Electronic Thesis and Dissertation Repository*. 9244.
<https://ir.lib.uwo.ca/etd/9244>

This Dissertation/Thesis is brought to you for free and open access by Scholarship@Western. It has been accepted for inclusion in Electronic Thesis and Dissertation Repository by an authorized administrator of Scholarship@Western. For more information, please contact wlsadmin@uwo.ca.

Abstract

Data-driven predictive maintenance for heat, ventilation, and air conditioning (HVAC) systems has gained much popularity over recent years due to the increasing availability of integrated internet of things (IoT) sensors capable of reporting HVAC internal operational data. Most existing predictive maintenance methods are designed to analyse these internal operational data for maintenance decision making. However, these methods are not applicable to HVAC systems that are not equipped with internal IoT sensors. Consequently, we propose an AutoEncoder and Artificial Neural Network based HVAC Health Prognostics framework (AE-ANN-HP) that classifies the health condition of HVAC systems using only daily power consumption and outside temperature readings, both of which are easy to obtain for non-IoT enabled HVAC systems. AE-ANN-HP when evaluated with three types of autoencoders all show an increase in performance when compared to existing HVAC health prognostics methods in terms of classification accuracy.

Keywords: data-driven predictive maintenance, health prognostics, condition monitoring, HVAC, smart building, machine learning.

Summary for Lay Audience

Heat, ventilation, and air conditioning (HVAC) systems account for a significant portion of energy consumption globally. Effective maintenance of HVAC systems can not only reduce the operational costs of the buildings but also improve human comfort of indoor environments. Data-driven predictive maintenance for HVAC systems has gained much popularity over recent years due to the increasing availability of integrated internet of things (IoT) sensors capable of reporting HVAC internal operational data. Most existing predictive maintenance methods are designed to analyse the internal operational data for maintenance decision making. However, these methods are not applicable to older or simpler HVAC systems that are not equipped with internal IoT sensors. Consequently, we propose an AutoEncoder and Artificial Neural Network based HVAC Health Prognostics framework (AE-ANN-HP) that classifies the health condition of HVAC system using only daily power consumption and outside temperature readings, both of which are easy to obtain for non-IoT enabled HVAC systems. The AE-ANN-HP framework is evaluated with three different types of autoencoders. Their performances are compared to existing HVAC health prognostics methods that are evaluated on the same dataset. The results show that all three AE-ANN-HP configurations yield higher classification accuracy than existing methods.

Acknowledgements

Above all, I would like to express my sincere gratitude to my supervisor, Dr. Miriam Capretz. Thank you for providing me with the opportunity to pursue a master degree despite me being an international student with elevated tuition fees. Thank you for also patiently guiding me through the obstacles in my research when I lost my direction and felt overwhelmed. It has been my pleasure to be your student.

I would like to thank my parents for your unconditional love and support. Because of you, I am privileged enough to have been studying abroad for ten years. Thank you for setting my life to easy mode.

I would like to thank my girlfriend for your company for the past five years. Thank you for all your emotional support when I struggled with my research and with the writing of this thesis. I would not have survived to this moment without your love and support.

I would like to also extend my gratefulness to all my friends in my research lab. Thank you Santiago for answering all my quick questions that almost always turned into hour long discussions. Thank you Luisa for your guidance during the early stage of my research while you were still in this lab. Thank you Patrick for chatting with me non stop and distracting me from work (just kidding). And thank you Dagimawi, Veronica, Amanda, Fadi, Madhushan, Mohammad, and the other Mohammad for keeping me company in the lab during my two years of life as a researcher.

Lastly, special thanks to Chiron and Enzo for cheering me up and doing silly things to make me laugh all day long. Perks of being cute little budgies.

Contents

Abstract	ii
Summary for Lay Audience	iii
Acknowledgements	iv
List of Figures	viii
List of Tables	x
List of Abbreviations, Symbols, and Nomenclature	xii
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	4
1.3 Thesis Outline	5
2 Background	6
2.1 Introduction	6
2.2 Artificial Neural Networks	6
2.3 Autoencoder	8
2.3.1 Convolutional Autoencoder	9
2.3.2 Recurrent Autoencoder with LSTM Cells	10
2.4 Building Energy Simulation with EnergyPlus	11
2.5 Summary	13

3	Related Works	14
3.1	Introduction	14
3.2	Data-driven HVAC Predictive Maintenance	14
3.2.1	HVAC Fault Detection and Diagnostics	15
3.2.2	HVAC Health Prognostics	16
3.3	Predictive Maintenance with Synthetic Data	18
3.4	Summary	19
4	AE-ANN-HP Framework	21
4.1	Introduction	21
4.2	Overview	21
4.3	Data Preprocessing	22
4.3.1	Data Sanitization	23
4.3.2	Normalization	25
4.3.3	Feature Engineering	25
4.4	Autoencoder Degradation Analysis	26
4.4.1	Autoencoder Model Training	28
4.4.2	Optimal Model Selection	28
4.4.3	Autoencoder Feature Generation	31
4.5	Health Condition Classification	32
4.6	Summary	32
5	Synthetic HVAC Data Generation with EnergyPlus	34
5.1	Introduction	34
5.2	Description of the Real Dataset	34
5.3	Synthetic Data Generation for HVAC system	36
5.4	Synthetic Data Generation for HVAC System with Degradation	38
5.5	Summary	39

6	AE-ANN-HP Framework Evaluation	40
6.1	Introduction	40
6.2	Dataset Split	40
6.3	Evaluation Metrics	42
6.4	Experiments & Results	44
6.4.1	Experimental Setup	45
6.4.2	Experiment I: Performance Baseline	45
6.4.3	Experiment II: AE-ANN-HP with Dense Autoencoder	49
6.4.4	Experiment III: AE-ANN-HP with Convolutional Autoencoder	52
6.4.5	Experiment IV: AE-ANN-HP with Long Short-Term Memory Autoencoder	55
6.5	Discussion	58
7	Conclusion & Future Work	61
7.1	Conclusion	61
7.2	Future Work	63
	Bibliography	65
	Curriculum Vitae	72

List of Figures

2.1	The structure of a perceptron.	7
2.2	An artificial neural network with 2 hidden layers.	7
2.3	A DAE with one hidden layer.	8
2.4	The structure of a typical convolutional autoencoder.	10
2.5	The structure of a LSTM autoencoder.	11
4.1	Proposed HVAC health prognostics framework.	23
4.2	Illustration of target classes and train, validation, and test set split.	26
4.3	Autoencoder model training, optimal model selection, and feature generation process.	27
5.1	HVAC power consumption and outside temperature trends from the real dataset: (a) From May to October, (b) In the first week of September.	35
5.2	HVAC power consumption and outside temperature trends from a synthetically generated dataset using EnergyPlus: (a) From May to October, (b) In the first week of September.	36
5.3	Comparison between the real dataset and the synthetic dataset: (a) Daily mean HVAC power consumption, (b) Pearson correlation coefficient of daily HVAC power consumption	37
5.4	Comparison of the HVAC power consumption in no degradation, moderate degradation, and severe degradation conditions.	39
6.1	Training, validation, and testing split of the dataset.	41

6.2	A multi-class confusion matrix. TP, TN, FP, FN are labeled in class C_1 's perspective.	42
6.3	ROC curves for a perfect classifier and a chance level classifier.	43
6.4	Comparison of confusion matrices of baseline models against the testing set: (a) ANN, (b) SVM, (c) DT. All values are averaged over three trials.	48
6.5	Comparison of confusion matrices against the testing set: (a) baseline ANN, (b) DAE-ANN-HP. Confusion matrix for baseline ANN is averaged over three trials. Confusion matrix for DAE-ANN-HP is averaged over nine trials.	52
6.6	Comparison of confusion matrices against the testing set: (a) baseline ANN, (b) CAE-ANN-HP. Confusion matrix for baseline ANN is averaged over three trials. Confusion matrix for CAE-ANN-HP is averaged over nine trials.	54
6.7	Comparison of confusion matrices against the testing set: (a) baseline ANN, (b) LSTMAE-ANN-HP. Confusion matrix for baseline ANN is averaged over three trials. Confusion matrix for LSTMAE-ANN-HP is averaged over nine trials.	57
6.8	Comparison of generalization performances between all 6 experimented methods: (a) F1 score, (b) AUROC score.	59

List of Tables

6.1	List of hyperparameters and their search space for tuning the baseline ANN model.	46
6.2	List of hyperparameters and their search space for tuning the baseline SVM model.	47
6.3	List of hyperparameters and their search space for tuning the baseline DT model.	47
6.4	Comparison of F1 and AUROC scores of baseline models against the testing set. All scores are averaged over three trials.	48
6.5	List of hyperparameters and their search space for tuning the DAE model. . . .	50
6.6	Hyperparameters of the five candidate DAE models.	51
6.7	Final health condition classification validation losses of DAE-ANN-HP.	51
6.8	Comparison of F1 and AUROC scores between DAE-ANN-HP and baseline ANN against the testing set. Scores for baseline ANN are averaged over three trials. Scores for DAE-ANN-HP are averaged over nine trials.	52
6.9	List of hyperparameters and their search space for tuning the CAE model. . . .	53
6.10	Hyperparameters of the five candidate CAE models.	54
6.11	Final health condition classification validation losses of CAE-ANN-HP.	54
6.12	Comparison of F1 and AUROC scores between CAE-ANN-HP and baseline ANN against the testing set. Scores for baseline ANN are averaged over three trials. Scores for CAE-ANN-HP are averaged over nine trials.	55
6.13	List of hyperparameters and their search space for tuning the LSTMAE model.	56
6.14	Hyperparameters of the five candidate LSTMAE models.	57

6.15 Final health condition classification validation losses of LSTMAE-ANN-HP. . . 57

6.16 Comparison of F1 and AUROC scores between LSTMAE-ANN-HP and baseline ANN against the testing set. Scores for baseline ANN are averaged over three trials. Scores for DAE-ANN-HP are averaged over nine trials. 57

List of Abbreviations, Symbols, and Nomenclature

AE-ANN-HP	Autoencoder and Artificial Neural Network based HVAC Health Prognostics
ANN	Artificial Neural Networks
AUROC	Area Under Receiver Operating Characteristic curve
BAS	Building Automation System
BIM	Building Information Modelling
CAE	Convolutional Autoencoder
CNN	Convolutional Neural Networks
DAE	Dense Autoencoder
DST	Daylight Saving Time
DT	Decision Tree
EPW	EnergyPlus Weather
FDD	Fault Detection and Diagnostics
FM	Facilities Management
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GBT	Gradient Boosted Tree
GMT	Greenwich Mean Time
HP	Health Prognostics
HVAC	Heat, Ventilation, and Air Conditioning
IDF	Input Data File
IoT	Internet of Things
IQR	Inter Quantile Range
LR	Learning Rate
LSTM	Long Short-Term Memory

LSTMAE	Long Short-Term Memory Autoencoder
MAE	Mean Absolute Error
MD	Moderate Degradation
MEP	Mechanical, Electrical, and Plumbing
MLE	Mean Logarithmic Error
MSE	Mean Squared Error
MSLE	Mean Squared Logarithmic Error
ND	No Degradation
NLP	Natural Language Processing
OVR	One-Vs-Rest
PdM	Predictive Maintenance
RE	Reconstruction Error
RF	Random Forest
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Networks
ROC	Receiver Operating Characteristic
RUL	Remaining Useful Life
SD	Severe Degradation
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
TMY	Typical Meteorological Year
TN	True Negative
TP	True Positive
TPR	True Positive Rate

Chapter 1

Introduction

1.1 Motivation

In 2021, around 135 Exajoules of energy was used in the operation of residential and commercial buildings worldwide. It accounted for 30% of global energy consumption and 27% of total greenhouse gas emissions [1]. Furthermore, 38% of buildings energy consumption come from the operations of heat, ventilation, and air conditioning (HVAC) systems [2]. The maintenance of these HVAC systems accounts for more than 65% of annual building facilities management (FM) costs [3]. Effective maintenance of HVAC systems can reduce maintenance costs, boost the availability of HVAC systems, and in turn improve human comfort of the indoor environments.

In facilities management, there are three main strategies toward maintenance management: corrective maintenance, preventive maintenance, and predictive maintenance [4, 5]. Corrective maintenance is the simplest form of maintenance strategy as it allows a piece of equipment to run to failure and only intervenes when breakdown happens. While this can be effective for non-critical equipment, i.e., those that do not cause disruption to the normal operation of the buildings, corrective maintenance is not ideal for HVAC systems because an unplanned breakdown can have a large impact on the operation of a building. Preventive maintenance is

a schedule-based maintenance strategy where an equipment is inspected and maintained regularly following a schedule derived from average-life statistics [4]. Traditionally, HVAC and many other systems in buildings are maintained following preventive maintenance strategy because it minimizes the chance of breakdown. However, the downside to preventive maintenance strategy is that an equipment can still have a lot of remaining useful life (RUL) at the time of a scheduled maintenance, which can create unnecessary waste. Additionally, frequent maintenance drives up FM costs and in turn increases the total operational cost of the buildings.

Predictive maintenance (PdM) strategy aims to solve the shortcomings of corrective and preventive maintenance strategies by actively monitoring and analysing the operational condition of an equipment for maintenance decision making. The concept of predictive maintenance has existed for more than 30 years. However, modern data-driven predictive maintenance is only recently popularized along with the adoption of an increasing number of Internet of Things (IoT) enabled equipment, which allow operational data to be collected and analysed at ease [6, 7]. There are two main approaches to data-driven predictive maintenance of HVAC systems, which are fault detection and diagnostics (FDD) and health prognostics (HP) [3]. Fault detection and diagnostics approaches aim to detect and classify incipient mechanical faults in the HVAC systems. On the other hand, health prognostics approaches aim to estimate the health status of the HVAC systems and predict their remaining useful life. Machine learning is commonly incorporated into both approaches, which involve training a machine learning algorithm with historical HVAC operational data and to make prediction of the HVAC system's status with real time data.

Artificial neural networks (ANN) are a type of supervised machine learning algorithm that has gained much popularity over recent years due to their ability to learn complex patterns from the data and make predictions with high accuracy. ANNs have been shown to perform well in both fault detection and diagnostics [8] and health prognostics [9] for HVAC systems. Autoencoders are unsupervised machine learning algorithms that are capable of learning a lower dimensional and generalized representation of the data. Because of their ability to produce

higher reconstruction errors for abnormal data samples than for normal ones, autoencoders have shown good performance in HVAC fault detection [10]. Although, autoencoders are rarely used in HVAC health prognostics approaches.

Nonetheless, due to the recent popularization of data-driven predictive maintenance, existing researches in the field of HVAC predictive maintenance has been mostly focused on more sophisticated HVAC systems capable of reporting a variety of internal operational data [9, 11, 12]. However, these methods are not applicable to simpler HVAC systems that are not IoT enabled and often lack comprehensive maintenance history, commonly found in residential and small commercial buildings. These systems account for a significant portion of the HVAC systems globally. Additionally, retrofitting internal IoT sensors to HVAC systems is often not feasible or desirable as it would introduce unnecessary wear and tear to the system. Consequently, in this thesis, we propose a data-driven predictive maintenance framework for health prognostics of HVAC systems that are unable to report internal operational status.

Previous research on HVAC power consumption patterns show that HVAC systems' power consumption is largely affected by the outside temperature trends with a correlation coefficient as high as 0.91 [13]. Furthermore, when working in degraded conditions, HVAC system's annual power consumption can increase by an average of 28.2% [14]. Hence, we aim to classify the daily health condition of an HVAC system using only its power consumption and the outside temperature trends, both of which are easily obtainable. HVAC power consumption data can be collected with the installation of a smart electric meter, while the outside temperature data can be collected with a smart temperature sensor or from a nearby weather station. Our proposed framework makes use of both an autoencoder and an ANN for daily health condition classification. The autoencoder model analyses the relative degradation of the HVAC system given the outside temperature trends and generates additional features to aid with the daily health condition classification by the ANN model.

1.2 Contributions

The main contributions of this thesis are summarized as follows:

- AutoEncoder and Artificial Neural Network based HVAC Health Prognostics (AE-ANN-HP) framework is a novel HVAC health prognostics solution designed for HVAC systems without internal IoT sensors. AE-ANN-HP incorporates an autoencoder to learn a generalized representation from daily HVAC power consumption and outside temperature data. The trained autoencoder then produces an enriched dataset which are used for daily HVAC health condition classification.
- A multi-objective fitness score is used for optimal autoencoder selection. This score assesses an autoencoder model's ability to minimize reconstruction errors for data samples in normal condition as well as its ability to discriminate between data samples from normal condition and various degraded conditions.
- Mean Logarithmic Error (MLE) function is proposed to calculate the difference between reconstruction errors for normal and degraded data samples. MLE function performs a natural logarithmic transform to the inputs, which suppresses the effects of larger differences and in turn boosts the effects of smaller differences.
- Show that using EnergyPlus, a building energy simulation software, to simulate HVAC behaviors under normal and various degraded conditions for synthetic data generation is a viable alternative to mitigate the lack of good quality data.
- Evaluation of AE-ANN-HP framework with three different autoencoder architectures. The performances are compared with existing health prognostics methods on four metrics which include confusion matrix, macro-averaged f1 score, macro-averaged OVR ROC curve, and AUROC score.

In all three experiments with different autoencoder architectures, the proposed AE-ANN-HP framework achieve better performance than existing HVAC Health Prognostics methods. AE-

ANN-HP with autoencoders more optimized for temporal feature extraction yield higher performance than AE-ANN-HP with a basic autoencoder.

1.3 Thesis Outline

The remainder of this thesis is organized as follows: Chapter 2 describes the background. It covers building energy simulation with EnergyPlus, artificial neural networks, autoencoders, and its variations.

Chapter 3 describes the related works. The two main approaches to data-driven predictive maintenance are discussed as well as predictive maintenance with synthetic data. The current research challenges and gaps are summarized at the end.

Chapter 4 presents the proposed AutoEncoder and Artificial Neural Network based HVAC Health Prognostics (AE-ANN-HP) framework including its three components.

Chapter 5 describes the synthetic data generation with EnergyPlus and the reason why it is a viable alternative for the real dataset.

Chapter 6 describes the experiments with AE-ANN-HP framework. It presents three experiments of the AE-ANN-HP framework with three different types of autoencoders and compares their performances with existing health prognostics methods. The experiment results and findings are discussed at the end.

Chapter 7 concludes this thesis and discusses future work.

Chapter 2

Background

2.1 Introduction

This chapter provides background knowledge for this work. We discuss machine learning algorithms including artificial neural networks and autoencoder and its variants. We also discuss building energy simulation with EnergyPlus.

2.2 Artificial Neural Networks

An artificial neural network (ANN) is a machine learning algorithm inspired by the biological brain. The building block of an ANN is a perceptron [15], depicted in Figure 2.1. A perceptron consists of an input vector \mathbf{x} , a weight vector \mathbf{w} , and an activation function Φ . A bias term b is also often included to adjust the y-intercept of the output [15]. The output h of the perceptron is the linear combination of the input vector \mathbf{x} and weight vector \mathbf{w} passed through the activation function Φ , given by (2.1).

$$h = \Phi(\mathbf{w}^\top \mathbf{x}) \quad (2.1)$$

An ANN contains an input layer, some hidden layers, and an output layer. Each layer consists of some neurons, and each layer of neurons interconnected to ones in the previous layer.

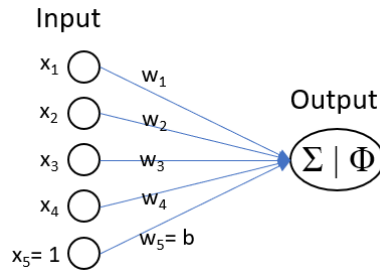


Figure 2.1: The structure of a perceptron.

Figure 2.2 shows an ANN with two hidden layers.

Training an ANN involves updating the weights in each neuron such that the prediction error, determined by an error function such as mean squared error (MSE) (2.2), where y is a vector of the ground truth and \hat{y} is a vector of model predictions, is minimized. The feature matrix is passed into the input layer of an ANN. The calculations for neuron outputs are then carried out for each hidden layer to produce the final output of the network for each feature vector. An error function calculates the error between network outputs and the target values. The back-propagation algorithm is leveraged to propagate the error back to in each hidden layer and calculate error gradient with respect to each weight in the network. Finally, the weights are updated by taking a step along the error gradient that leads toward a lower loss value, an algorithm called gradient descent. A learning rate parameter is often desired to control the size

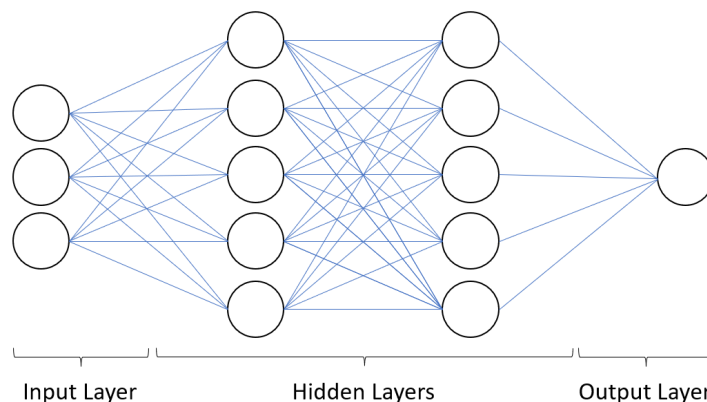


Figure 2.2: An artificial neural network with 2 hidden layers.

of the step for gradient descent.

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.2)$$

In this thesis, an ANN is used for multi-class HVAC health condition classification. The number of neurons in the output layer corresponds to the number of classes for the classification task such that the neurons represent the probabilities of the output classes. Additionally, softmax activation function is used for the output layer, which gives the probability distribution over mutually exclusive output classes. The weights in the network are adjusted to minimize the cross-entropy error function [16].

2.3 Autoencoder

In supervised learning, a machine learning model learns from a labeled dataset, i.e., the target output associated with each input is provided to train the model. In comparison, unsupervised machine learning models attempt to learn useful representation of the data without the need of labeled data. Autoencoders are unsupervised machine learning algorithms that attempt to

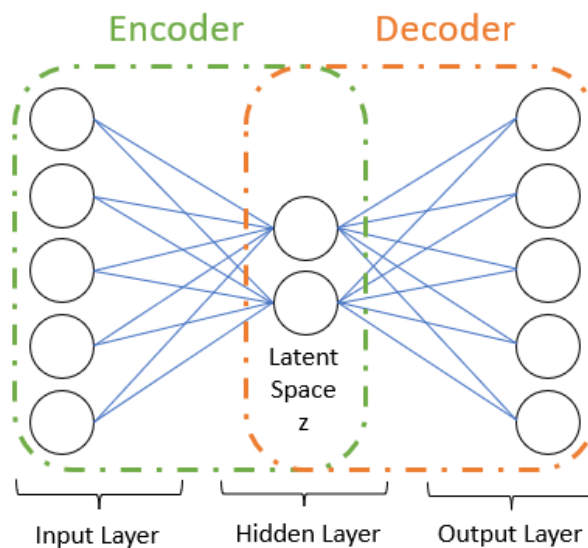


Figure 2.3: A DAE with one hidden layer.

copy its input to its output [17] and extract lower dimensional and generalized features through the process. They are commonly used for fault detection [18, 19, 20] based on the fact that autoencoders are capable of producing higher error values for faulty data samples than for normal ones. In this thesis, we use an autoencoder to analyze the degradation of HVAC system using its power consumption and outside temperature data.

A basic autoencoder is one consisting of densely connected layers, also known as dense autoencoder (DAE). Figure 2.3 depicts the structure of a DAE. Conceptually, an autoencoder contains an encoder and a decoder. The encoder part of an autoencoder transforms the input vector into a latent space z , usually of a smaller dimension [17]. The decoder part, on the other hand, reconstructs the input by transforming the latent space vector back to its original dimension. Similar to the prediction error in ANN, the reconstruction error (RE) of an autoencoder model is calculated by comparing the reconstructed input against the actual input.

Although a basic DAE can learn a compressed representation of the input features, it is not optimized to extract temporal relations within the data. Fortunately, two other autoencoder architectures, convolutional autoencoder [21] and recurrent autoencoder [22], have been proposed to overcome this problem.

2.3.1 Convolutional Autoencoder

Convolutional neural networks (CNN) apply convolution operations to only small local regions of the features to extract the spatial relationship. CNNs with 1D convolution have been shown to perform well with time-series inputs [23, 24], while CNNs with 2D convolution have seen great success with computer vision based problems such as image classification/segmentation [25, 26] and fault detection [18]. A convolutional autoencoder (CAE), first introduced by Masci *et al.* [21], is a combination of a CNN and an autoencoder. In addition to dense layers in a DAE, a typical CAE also includes series of convolution, maxpooling, upsampling, and deconvolution layers, as depicted in Figure 2.4.

The encoder of a CAE contains several pairs of convolution and maxpooling layers. Con-

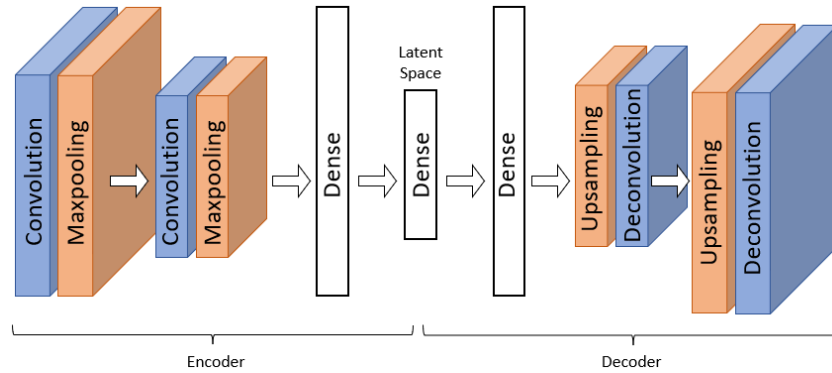


Figure 2.4: The structure of a typical convolutional autoencoder.

convolution layers apply convolution operation to the inputs with some filter to construct feature maps. Maxpooling layers help to achieve invariance to small translations of the input [17] as well as reducing the computational complexity. The stacking of convolution and maxpooling layers allows for the extraction of multiple levels of representations [27]. Finally, dense layers are used to further reduce the dimension. In the decoder part of a CAE, the effects of convolution layers and maxpooling layers are reversed by the use of upsampling layers and deconvolution layers. The same number of dense layers and upsampling and deconvolution layer pairs are applied to transform the latent space back into its original dimensions.

2.3.2 Recurrent Autoencoder with LSTM Cells

Conventional ANNs make the assumption that the data are independent in the time domain. In contrast, recurrent neural networks (RNN) model the time domain by including feedback loops in RNN cells to learn from sequences of data. In addition to weights, RNN cells contain hidden states to memorize time dependent information. Long Short-Term Memory (LSTM) cells are a special kind of RNN cells designed to mitigate the vanishing and exploding gradient problems [28] experienced during the training of traditional RNNs. Because of its ability to discover temporal relationships, LSTM networks have seen great success in problems involving temporal sensitive data, including natural language processing (NLP) [29] and time series forecasting [30].

In an LSTM autoencoder (LSTMAE), LSTM layers, each containing some LSTM cells, are used to encode and decode the time series features [22]. This thesis uses the LSTM autoencoder structure proposed by Sagheer *et al.* [31], as shown in Figure 2.5. It is unfolded along the time axis, which contains 3 time steps. This LSTMAE model has one LSTM layer in the encoder and another one in the decoder. In the encoder part of the model, the LSTM layer accepts input features from each time step and outputs the latent space vector after the last time step. In the decoder part of the model, the latent space vector is repeated and passed as input to the LSTM layer during each time step. This layer returns the full output sequence, i.e., one output for each time step. At the end, a dense layer is applied to the output in each time step to reduce the output dimension from the number of LSTM cells to the actual input dimension.

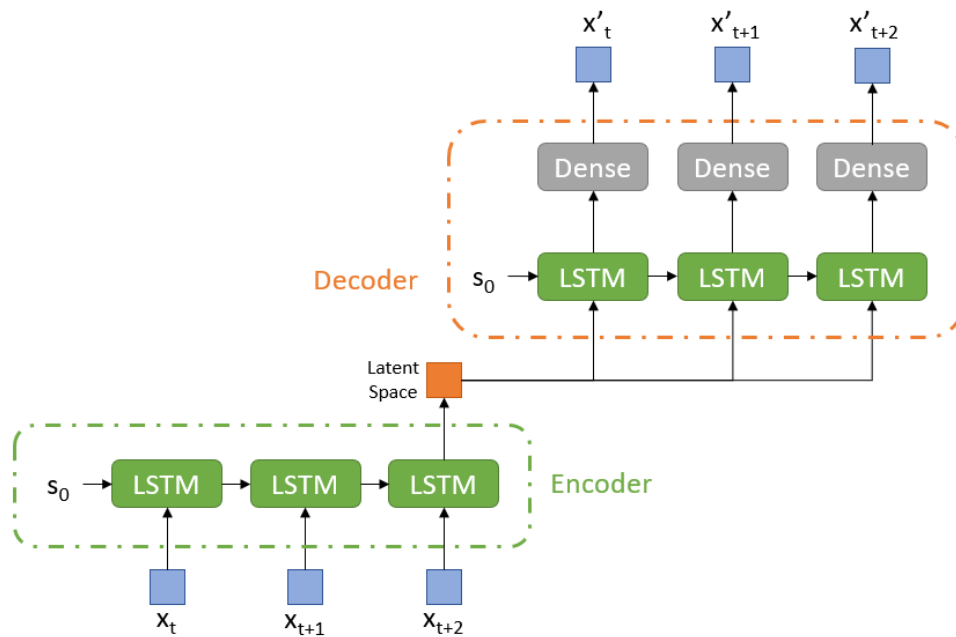


Figure 2.5: The structure of a LSTM autoencoder.

2.4 Building Energy Simulation with EnergyPlus

EnergyPlus [32] is an open-source building energy simulation software whose development is funded by the U.S. Department of Energy. It is capable of simulating the energy consump-

tion for including but not limited to heating, cooling, ventilation, and lighting of the building. EnergyPlus has been used by researchers in many field of studies including building thermal management [33], potential energy savings [34], and predictive maintenance [35]. In this thesis, we utilize EnergyPlus to generate synthetic HVAC power consumption data in both normal operation condition and several degraded conditions for evaluating the proposed HVAC health prognostics framework. To run a simulation, two main components need to be prepared, which include building modeling component and weather information component.

To model a building in EnergyPlus, an IDF (Input Data File) file needs to be constructed. The IDF file contains information about the location/exterior environment of the building, its construction/building envelope, and the HVAC system installed. The modeling of building construction is achieved by defining the zoning, surface geometry, and construction materials. Various types of HVAC systems can also be modeled. Because of the complexity of building modeling, EnergyPlus provides a number of predefined IDF files, which covers different combinations of building structures and HVAC systems, for use by researchers. In this thesis, we adopt and modify a predefined IDF file named “5ZoneAirCooled”.

Weather information is also required for EnergyPlus to simulate building energy. EnergyPlus uses a proprietary format named EPW (EnergyPlus Weather) for weather data storage. The EPW file contains annual hourly weather information such as dry bulb/wet bulb temperature, dew point temperature, wind direction and speed, and solar radiation. While it is possible to construct custom EPW files, weather features such as solar radiation and horizontal diffuse radiation are usually harder to acquire than other simpler features commonly available through weather stations. For this reason, EnergyPlus and some other 3rd parties provide EPW files for some typical meteorological years (TMY) for many locations around the world for building energy simulation.

In addition to building energy simulation, EnergyPlus also has native fault models [36] that allows for energy simulation under some faulty conditions. Some fault types include thermostat/humidistat offset, heating and cooling coil fouling, and dirty air filters. For each fault

model, an intensity parameter can be modified to vary the intensity of the fault. In this thesis, the HVAC power consumption under degraded condition is simulated with native fault models.

2.5 Summary

In this chapter, we discuss machine learning algorithms used in this thesis including ANN, DAE, CAE, and LSTMAE. Our proposed framework consists of an autoencoder for HVAC degradation analysis and an ANN for final daily HVAC health condition classification. The proposed framework is designed to work with an autoencoder of any type. In this thesis, we evaluate the performance of our proposed framework with the use of DAE, CAE, and LSTMAE. We also introduce building energy simulation with EnergyPlus. In this thesis, EnergyPlus is used to generate synthetic HVAC dataset for use in experiments with the proposed framework.

Chapter 3

Related Works

3.1 Introduction

This chapter reviews data-driven HVAC predictive maintenance methods and data augmentation and synthetic data generation techniques. Research challenges and opportunities are identified.

3.2 Data-driven HVAC Predictive Maintenance

The study of data-driven predictive maintenance gained much interest from researchers in the last decade through the fourth industrial revolution, also known as industry 4.0, where the mass adoption of sensors on industrial equipment allows for unprecedented amount of data to be collected and analysed for maintenance decision making [6, 7]. The current state-of-the-art methods for data-driven predictive maintenance for HVAC systems are reviewed. Two most popular predictive maintenance approaches, namely fault detection and diagnostics (FDD) and health prognostics (HP), and various machine learning algorithms being applied are discussed.

3.2.1 HVAC Fault Detection and Diagnostics

Fault detection and diagnostics for HVAC systems refers to the early detection of incipient faults and the isolation of fault type before the total breakdown of an HVAC system happens. Several machine learning algorithms have been applied to HVAC Fault Detection and Diagnostics, which include: SVM [8, 37], decision tree [37, 38, 39, 40], ANN [8], and RNN [8, 10, 11].

Satta *et al.* [38] proposed a dissimilarity-based approach in which fault detection is accomplished by investigating the mutual dissimilarities of multiple HVAC systems within a cohort. In their method, the authors train a gradient boosted tree (GBT) binary classifier with dissimilarity values to detect HVAC faults and found their method to outperform other methods that only consider individual HVAC systems. However, because the HVAC systems need to be mechanically identical and operate in the same conditions, their method has limited application in cases where different kinds of HVAC systems or only a single HVAC system is available for study.

Bouabdallaoui *et al.* [10] developed a building facility predictive maintenance framework that includes five steps, namely data collection, data processing, model development, model feedback and improvement, and model deployment. In their framework, the authors propose to use an LSTM autoencoder architecture with data collected from building automation system (BAS) and vibration sensors installed onto equipment's surface for fault detection. Specifically, a fault is detected when the root mean square error (RMSE) reconstruction error surpasses an end-user defined threshold. Their framework is evaluated on HVAC installations in a sport facility building. However, the authors admit the small data collection period and the need of more extensive evaluation.

Taheri *et al.* [11] investigated 7 different configurations of LSTM networks and their abilities to detect HVAC faults and classify the types of fault, including stuck outside air damper, stuck heating coil valve, stuck cooling coil valve, and outdoor air temperature bias. They considered LSTM networks including standard LSTM networks, deep transition input LSTM networks, deep transition output LSTM networks, stacked LSTM networks, and combinations

of such. The authors evaluated the models on HVAC internal sensor data and found that stacked deep transition output deep recurrent neural network configuration yielded a f1-score of 0.91 on their Single-zone Air Handler Unit dataset, outperforming the random forest (RF) model by 6% and the gradient boosted tree (GBT) model by 11%.

Tasfi *et al.* [41] proposed a semi-supervised machine learning technique to achieve anomaly detection of building systems including HVAC systems using electrical data reported by IoT electric meters. The authors used a convolutional autoencoder with the addition of a classifier that takes input from the latent space of the autoencoder to detect anomaly in the electrical signal. Our proposed framework is similar to their method in that both methods incorporate an autoencoder and a classifier. In contrast, our proposed framework classifies the macroscopic degradation of the HVAC system instead of microscopic anomalies in the electrical signal. In addition, the autoencoder used in their method is intended to learn an embedding, whereas the autoencoder in our proposed framework is designed to not only learn the embedding of the data but also to differentiate between data samples in normal and degraded health conditions through separations of reconstruction errors.

These studies show that the fault detection and diagnostics is a promising approach to data-driven predictive maintenance for HVAC systems. However, the FDD approaches do not provide a macroscopic view of the degradation trends, which is crucial to understanding the degradation dynamics for the HVAC systems. Additionally, the majority of studies in this field require IoT sensors embedded into the HVAC systems for operational data collection. Hence, they are not applicable to the HVAC systems that are not IoT enabled or do not collect operational data at all. In contrast, our approach focuses on the health prognostics of HVAC system and aims to provide a solution for the HVAC systems without internal IoT sensors.

3.2.2 HVAC Health Prognostics

Health prognostics for HVAC systems refers to the assessment of the operational condition of HVAC systems for predictive maintenance planning. Health prognostics include the predic-

tion of remaining useful life (RUL) and the estimation of health condition of HVAC systems. Health prognostics differ from fault detection and diagnostics in that FDD are generally binary classification in the case of fault detection, or multi-class classification in the case of fault diagnostics; whereas in most cases, health prognostics are regression problems. HVAC health prognostics is also less popular than HVAC fault detection and diagnostics in terms of the number of studies done on the topics. Nonetheless, several machine learning algorithms have been applied by researchers for HVAC health prognostics, including ANN [9], SVM [9], and decision tree [12].

Cheng *et al.* [9] proposed a predictive maintenance planning framework for mechanical, electrical, and plumbing (MEP) components. Their framework contains the information layer, which collects and integrates data from building information modelling (BIM) sensor network, facilities management (FM), and application layer, which is responsible for fault alarming and condition prediction. The authors proposed the idea of condition index, a value between 0 and 10, that represents the health status of the component where higher number indicates better condition. The authors experimented with ANN and SVM models with sensor data and FM data to predict the condition index and found the SVM model to yield slightly higher performance when tested against HVAC chillers. However, their method requires manual data labeling from FM experts. This can be very costly and not be feasible in some cases.

Yang *et al.* [12] proposed to use the operation data leading up to each HVAC failures for the remaining useful life (RUL) prediction of HVAC components. Sensor and actuator data up to two weeks before the occurrence of a failure are used to predict the RUL of the component in number of days. To evaluate their method, the authors extracted data from 11 failure events and applied regression tree model for the task. They found their method can reliably predict the RUL in 5 out of 11 test cases, while in the other 6 test cases, their method yielded slightly higher regression errors. Their method mitigates the need of data labelling. However, because the predicted RUL has a small range of 14 days, it does not provide health prognostics from a macroscopic view which would capture the gradual degradation of the HVAC systems. The

authors also reported that some failures may not be detectable from data trends leading up to the failure.

These studies show the promising potential of health prognostics approach for HVAC predictive maintenance. Both studies frame the HVAC health prognostics as regression problems. In contrast, our approach attempts to assign HVAC health conditions into classes and rely on synthetic data generation techniques to generate data samples for each health condition. Our approach eliminates the need for manual labeling while still focusing on the macroscopic level HVAC health prognostics. Additionally, similar to the studies on fault detection and diagnostics, both studies still require extensive operational data collection efforts. In contrast, our proposed method provides a solution to HVAC predictive maintenance for HVAC systems that are not capable of reporting internal operational status or cannot be easily converted to do so.

3.3 Predictive Maintenance with Synthetic Data

Researchers in the field of data-driven predictive maintenance often face the problem of a lack of high quality data. For fault detection and diagnostics approaches, the dataset are often highly imbalanced, e.g., the majority of the samples are negative samples. For health prognostics approaches, run-to-failure dataset are often unobtainable because HVAC systems are considered critical components, i.e., most follow preventive maintenance strategy. To mitigate this problem, several data augmentation and synthetic data generation techniques have been applied to create synthetic dataset that aid the research.

Ebrahimifakhar *et al.* [37] employed a physics-based model [42, 43] to generate a synthetic dataset containing various HVAC faults for fault detection and diagnostics. However, this introduces the problem of imbalanced dataset because while faulty classes can be generated with different intensities, the no fault class cannot. The synthetic minority oversampling technique (SMOTE) [44] was then applied to augment the no fault class to produce a more balanced dataset. The authors found that by augmenting the no fault class, their classifier suffers

a slight decrease in overall accuracy but yields significantly higher performance for no fault class prediction. Galvez *et al.* [39] created a physics-based model in MATLAB for an HVAC system installed in passenger train carriages. Their model contains real sensors located in the real HVAC system and virtual sensors that simulate outputs based on data from the real sensors. The synthetic faulty data is generated through introducing a drift to the virtual sensor outputs. The authors also introduced noise in input parameters for augmenting the dataset.

The synthetic data generation techniques discussed above focus on the simulation of the physical internal workings of the HVAC system for synthetic sensor data generation. On the other hand, several studies have also attempted to generate synthetic data by the use of EnergyPlus, an building energy simulation software. Yan *et al.* [45] generated synthetic data by varying parameters such as chiller tube inside diameter and fan efficiency in the definition of HVAC system and subsystems in EnergyPlus. Chakraborty *et al.* [35] created two office building models in EnergyPlus and utilised the integrated fault models [46] to introduce faulty behaviors for generating synthetic data. In this thesis, we combine the techniques from these two studies, i.e., varying the parameters in EnergyPlus fault models, such that faulty behaviors with multiple intensities are generated.

3.4 Summary

In this chapter, the two data-driven HVAC predictive maintenance approaches, namely fault detection and diagnostics and health prognostics, as well as synthetic data generation and data augmentation techniques are reviewed. The current research challenges and gaps are summarised as follows:

- Most studies in the field focus on HVAC systems equipped with internal IoT sensors such that the internal operational conditions can be monitored for analysis. While they can be beneficial to more modern and complex HVAC systems, these methods are not suitable for those systems without IoT internal sensors. Our proposed method aims to fill in this

gap by estimating the health condition of HVAC systems using power consumption and weather data, both of which are easier to collect.

- While HVAC fault detection and diagnostics has been well studied, HVAC health prognostics has not seen many advancements. Our proposed method attempts to study the degradation of HVAC systems from a macroscopic standpoint such that the deterioration process is captured.

Chapter 4

AE-ANN-HP Framework

4.1 Introduction

This chapter describes the proposed AutoEncoder and Artificial Neural Network based HVAC Health Prognostics (AE-ANN-HP) framework, a novel data-driven HVAC health prognostics framework using HVAC power consumption and weather data. An overview of the framework is given. Then, subsequent sections describe the framework components in detail.

4.2 Overview

The proposed AE-ANN-HP framework is shown in figure 4.1. This framework consists of three major components: data preprocessing, autoencoder degradation analysis, and health condition classification. The purpose of data preprocessing component is to sanitize the data as well as to engineer more features. Because of the limited features available to non-IoT enabled HVAC systems, it is important to extract more features to build a richer dataset for analysis. The autoencoder degradation analysis component is designed to differentiate between HVAC behaviors in normal and degraded conditions. An autoencoder is used for this component because it is capable of producing higher reconstruction errors for a data sample in degraded conditions relative to if it is in normal condition. We also propose a novel multi-objective

fitness function for hyperparameter optimization of the autoencoder. The latent space of the autoencoder and the reconstruction errors are then used to enrich the dataset. Because of the dynamic nature of HVAC behaviors on a daily basis, an HVAC system in normal operating condition may consume more power in a hotter day than the same HVAC system in a degraded condition would in a cooler day. This means that the autoencoder degradation analysis component can only analyse the relative degradation of the HVAC system and it alone is not enough to accurately classify its final health condition. Hence, the health condition classification component handles the final classification of HVAC daily health condition with an artificial neural network (ANN) multi-class classifier. Because this research focus on the health prognostics aspect of predictive maintenance and the mechanical degradation of the HVAC system is a slow process, AE-ANN-HP framework analyses the daily HVAC power consumption and outside temperature readings to predict the daily health conditions.

The flow of data through the framework is described as follows. Beginning with raw data, i.e., the HVAC power consumption and outside temperature time series data, the data preprocessing component prepares the data and splits the data into time-series features and non-time series features. Then, the autoencoder degradation analysis component takes in the time-series features and produces the latent space representation of the features as well as the reconstruction errors. Finally, these two outputs are concatenated with the non-time series features and fed into the health condition classification component to produce a final daily health condition for the HVAC system.

4.3 Data Preprocessing

The first component in the proposed HVAC health prognostics framework is data preprocessing. This component is necessary to transform raw data that often contain impurities into a better form that is suitable for machine learning models to train on. This component is further divided into three sub-components, namely data sanitization, normalization, and feature engi-

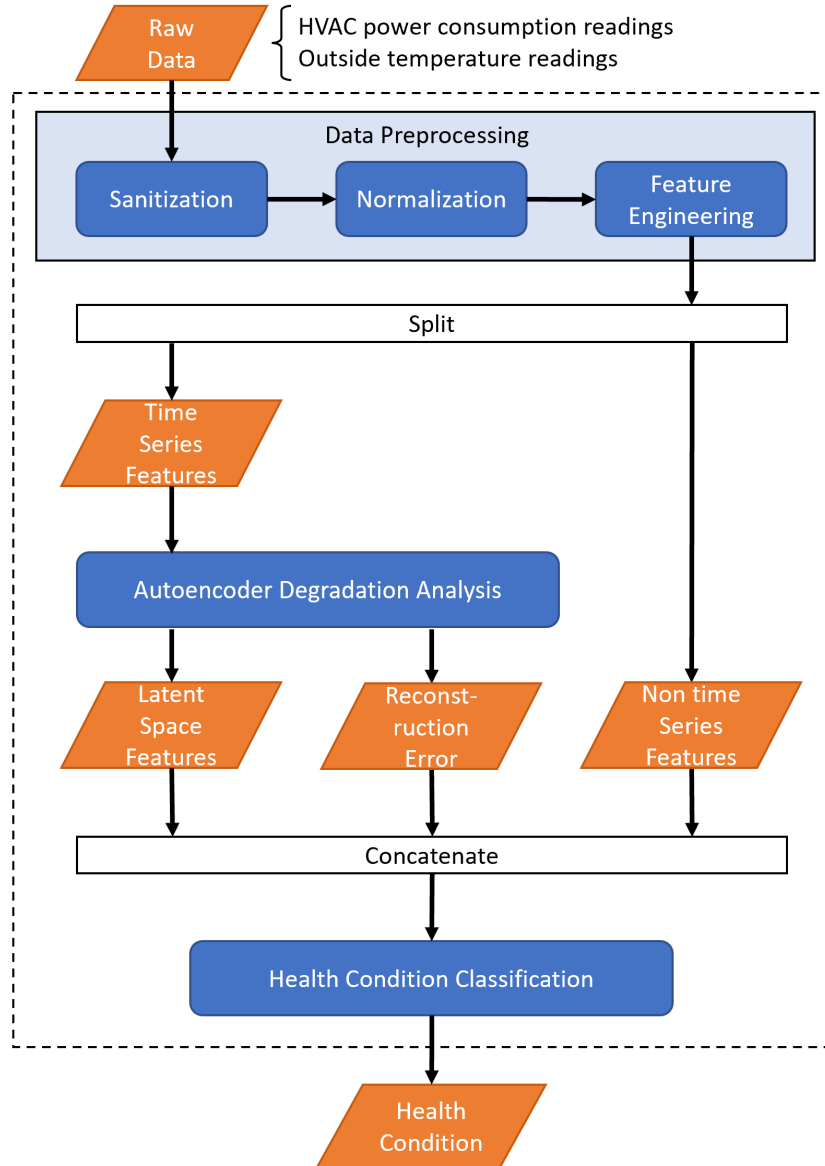


Figure 4.1: Proposed HVAC health prognostics framework.

neering. Data sanitization is needed because raw data often contain impurities which need to be removed. Normalization scales all data into the same range for machine learning algorithms to learn more effectively. Feature engineering is also beneficial as it helps build a richer dataset.

4.3.1 Data Sanitization

The data sanitization process in this framework is done in three steps. First, because missing values are commonly found in time series data, the missing data replacement step is used to

mitigate this problem. Second, in parts of the world which implement daylight saving time (DST), the artificial 1 hour shift in time during summer can cause the HVAC behavior to deviate, hence daylight saving time adjustment step is needed to correct the shift in HVAC behavior. Finally, some HVAC systems have regular inactivity periods. Inactivity period removal step is used to remove these periods. Each step is explained in more detail as follows.

- ***Missing Data Replacement*** Time series data such as HVAC power consumption and outside temperature collected by IoT smart electric meters and temperature sensors often contain missing values due to temporary loss of communication of the smart meters. To mitigate this issue, missing observations are replaced with the average of the previous and the next valid observations.
- ***Daylight Saving Time Adjustment*** When daylight saving time is in effect, time series data readings reported in Greenwich mean time (GMT) would become out of sync with local time by 1 hour because of the shift. Hence, the timestamps on these observations are rolled back 1 hour such that they align with local time. Then, the skipped 2 A.M. during a transition into daylight saving time is filled according to the aforementioned missing data replacement method. On the other hand, the duplicated 2 A.M. during a transition back to standard time is removed.
- ***Inactivity Period Removal*** The HVAC systems installed in commercial locations are typically setback (set to a more energy conserving temperature) or shutdown during unoccupied hours to conserve energy [47]. Because the observations made during these times do not contain any useful information, removing these readings can reduce the amount of trainable parameters in the model with no impact to the performance. This process can be ignored for HVAC systems installed in residential locations where the corresponding building(s) may be occupied at any given time.

4.3.2 Normalization

Data normalization is critical in machine learning because features with a larger scale can overshadow those with a smaller scale, resulting in inferior performances [48]. In this thesis, we adopt min-max scaling (4.1) to normalize the time series data because Min-Max scaler preserves the relationships among the input data [49], which is beneficial for an autoencoder in the autoencoder degradation analysis component to effectively learn the generalized representation of the data. In (4.1), \mathbf{x}_i is a value in the dataset, \mathbf{x}_{min} and \mathbf{x}_{max} are the minimum and maximum values of the dataset, and \mathbf{x}_i' is the normalized value.

$$\mathbf{x}_i' = \frac{\mathbf{x}_i - \mathbf{x}_{min}}{\mathbf{x}_{max} - \mathbf{x}_{min}} \quad (4.1)$$

Because data samples containing degraded HVAC behaviors might not always be available, the minimum and maximum values for the scaler are determined based only on data samples with no degradation. The normalization transformation is then applied to all samples in the dataset.

4.3.3 Feature Engineering

Feature engineering is important especially in this case where the available features are limited. To engineer additional features from the time-series data, the data is first grouped into daily windows such that the data in each window contain timestamps for one calendar day. Then, statistical features are calculated for each window. In this thesis, we adopt daily minimum, daily maximum, daily mean, and daily standard deviation for statistical features. Additionally, in some settings, HVAC operation can be influenced by date and time, e.g., different operational profiles for different days in a week, or in different seasons. Hence, date related features are extracted based on the date of each window. This can include month of the year, day of the year, and day of the week. To preserve the periodical nature of date related features, these features are encoded with trigonometric encoder, given by:

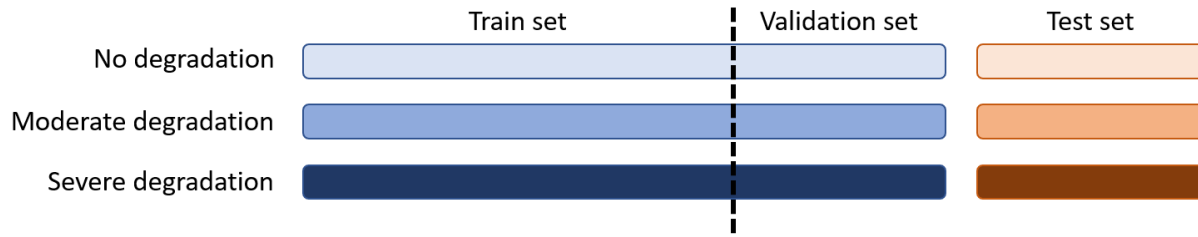


Figure 4.2: Illustration of target classes and train, validation, and test set split.

$$\mathbf{x}_i^{sin} = \sin\left(\frac{2 \cdot \pi \cdot \mathbf{x}_i}{\tau}\right) \quad (4.2a)$$

$$\mathbf{x}_i^{cos} = \cos\left(\frac{2 \cdot \pi \cdot \mathbf{x}_i}{\tau}\right), \quad (4.2b)$$

where τ is the period, e.g., 7 days in a week. Lastly, the target classes are generated. In this thesis, we use three target classes for determining the daily health condition of HVAC systems. They are ND for no degradation condition, MD for moderate degradation condition, and SD for severe degradation condition. These three target classes are based on the finding that HVAC systems operating in moderately degraded condition and severely degraded condition show increases in annual power consumption of 14.3% and 28.2% [14]. However, this framework is not restricted to work with only three health condition classes such that more classes may be considered. Figure 4.2 demonstrates the three target classes as well as the train, validation, and test set split of the dataset.

4.4 Autoencoder Degradation Analysis

The second component of the framework is the degradation analysis component. This component incorporates an autoencoder to differentiate between HVAC behaviors in normal and degraded conditions as well as generate additional features to aid the final classification of daily health conditions. In this component, an autoencoder is applied to the time series data in each daily window to analyse their relative degradation. The autoencoder model is trained only on no degradation data samples. Because the model only sees data samples from no degrada-

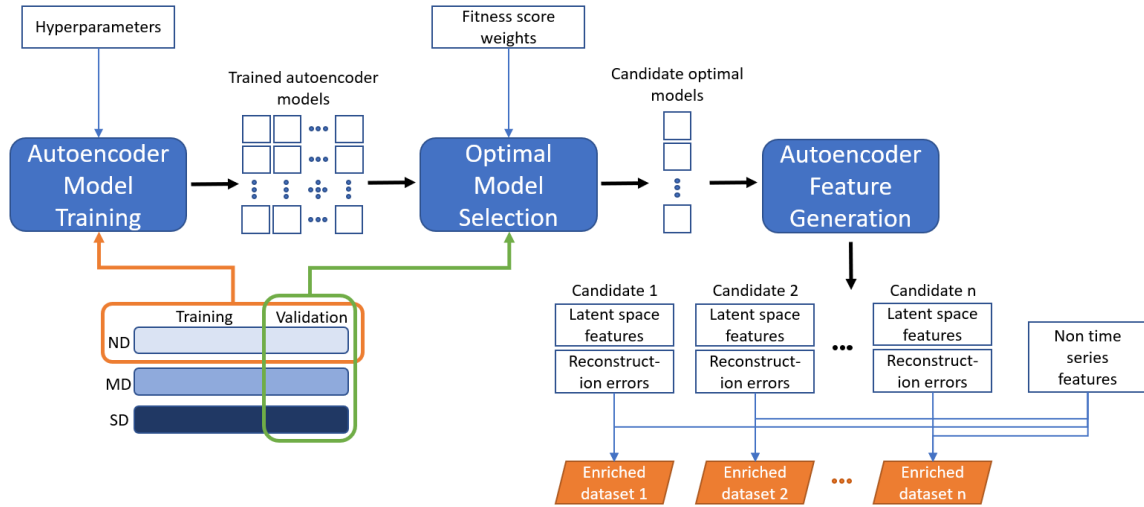


Figure 4.3: Autoencoder model training, optimal model selection, and feature generation process.

tion class during training, its parameters are optimized such that the reconstruction errors are minimized for no degradation samples. Hence, the model is expected to produce higher reconstruction errors for samples belonging to moderate degradation and severe degradation classes when similar outside conditions are exhibited. Once the autoencoder model is trained, the encoder part of the model is capable of compressing the input data into a lower dimensional representation with minimal loss of information, also known as the latent space. Hence, both the reconstruction error and latent space features are used to generate an enriched dataset to be used in the health condition classification component.

Because the autoencoder is used to generate an enriched dataset for the final health condition classification, the hyperparameter for the autoencoder model needs to be tuned so that the best performing model is selected for this purpose. Hence, the selection of the optimal autoencoder model is incorporated into autoencoder model training. This process adopts the exhaustive grid search strategy for model training and finds a list of candidate optimal autoencoder models based on their fitness scores. Hence, the autoencoder degradation analysis component is split into three steps: autoencoder model training, optimal model selection, and autoencoder feature generation, as shown in Figure 4.3. These steps are detailed in the following three subsections.

4.4.1 Autoencoder Model Training

The first step of the process is autoencoder model training. In this step, the autoencoder model training with all hyperparameter sets are conducted. First, the hyperparameters for the desired autoencoder model are identified. Let's assume the use of a dense autoencoder (DAE). Common hyperparameters used for tuning include the number of hidden layers, the number of neurons in each hidden layer, activation function, and learning rate [50]. Exhaustive grid search strategy is then used such that an autoencoder model is trained for each possible combination of the hyperparameter values. The autoencoder models are trained with no degradation training set, while the no degradation validation set is used for early stopping of the training process to prevent model over-fitting [51]. Fitness scores are calculated for these models to determine the list of candidate optimal models in the next part of the process.

4.4.2 Optimal Model Selection

After all the autoencoder models are trained, a list of candidate optimal models are determined by the calculation of fitness scores. Conventionally for autoencoder hyperparameter optimization, the fitness of a model is usually determined using reconstruction errors [52, 53], i.e., the model that minimize the reconstruction errors for a validation dataset is the optimal model. However, in our scenario, minimizing the reconstruction error alone is not enough to determine the optimal model. To generate the best features for health condition classification, the autoencoder needs to not only minimize the reconstruction error for no degradation data samples, but also distinguish between no degradation, moderate degradation, and severe degradation samples in terms of the separation of reconstruction errors for each class. Hence, we propose a multi-objective optimization function to find the optimal model.

The first objective is to minimize the reconstruction errors for the no degradation validation set. Specifically, the mean of reconstruction errors (4.3) is minimized. RE is a vector of

reconstruction errors and \overline{RE} is the mean of reconstruction errors.

$$\overline{RE} = \frac{1}{n} \sum_{i=1}^n RE_i \quad (4.3)$$

The second objective is to maximize the separation between the reconstruction errors for each of the classes. Conventional error functions such as mean squared error (MSE) and mean absolute error (MAE) are not ideal in this scenario because these functions are designed such that the signs of the errors are ignored. Because of the inclusion of square operation or absolute value operation, the errors are always positive. This can result in an autoencoder model that fails to separate the three classes, e.g., its reconstruction errors of no degradation class are higher than those of degraded classes, yielding a larger separation score than another autoencoder model which is capable of separating the three classes but only by a small amount. Additionally, because the HVAC power consumption trends vary from day to day, depending largely on the daily outside temperature trends [13], the differences between reconstruction errors for each data sample within the same class can also vary by a large amount. In this case, an ideal function to calculate the separation of reconstruction errors between each class is one that keeps the sign of the separation values and prevents larger separations from overshadowing smaller separations. Based on these desired characteristics, we propose the mean logarithmic error (MLE) function as the objective function for calculating the separations. The MLE function is defined by

$$MLE(RE_1, RE_2) = \frac{1}{n} \sum_{i=1}^n (\log_e(RE_{1i} + 1) - \log_e(RE_{2i} + 1)), \quad (4.4)$$

where RE_1 and RE_2 are two vectors of reconstruction errors. Note that the translation by 1 allows the natural log function to pass through the origin (0, 0). MLE applies natural log transformation to the two input vectors, then calculates the mean of the differences between the two transformed vectors. The MLE function is based on the existing Mean Squared Logarithmic Error (MSLE) function, which is designed to penalize under-estimated predictions

more harshly than over-estimated predictions. We remove the square operation from MSLE function to retain the signs of the outputs. We also assign the reconstruction errors vector of a more degraded condition to RE_1 and that of a less degraded condition to RE_2 such that positive separations (under-estimated predictions) yield higher positive scores than negative scores yielded by negative separations (over-estimated predictions). Overall, the MLE function not only maintains the sign of the separations but also drastically reduces the effects of larger differences by the log transform, which in turn enhances the effects of smaller differences. It prevents large differences from dominating all else. Hence, it is the ideal metric for determining the autoencoder models' ability to separate the reconstruction errors for the three classes.

Concretely, the two objective functions applied to the optimal autoencoder model selection process are shown by (4.5a) and (4.5b).

$$obj_1(RE_{ND_val}) = -\overline{RE}_{ND_val} \quad (4.5a)$$

$$obj_2(RE_{ND_val}, RE_{MD_val}, RE_{SD_val}) = \\ MLE(RE_{SD_val}, RE_{MD_val}) + MLE(RE_{MD_val}, RE_{ND_val}) \quad (4.5b)$$

For obj_1 , the mean of the reconstruction errors for the no degradation validation set, denoted as RE_{ND_val} , is calculated. Because this objective function is to be minimized, a negative sign is added in the front. For obj_2 , the MLE between reconstruction errors for severe degradation validation set and reconstruction errors for moderate degradation validation set and the MLE between reconstruction errors for moderate degradation validation set and reconstruction errors for no degradation validation set are added together. The reconstruction errors of severe degradation validation set, moderate degradation validation set, and no degradation validation set are denoted as RE_{SD_val} , RE_{MD_val} , and RE_{ND_val} , respectively. This function is to be maximized. After obtaining the results for both objective functions, a multi-objective fitness score is used

for assessing the fitness of each trained autoencoder model. It is defined as

$$fitness = w_1 \cdot |obj_1(RE_{ND_val})| + w_2 \cdot |obj_2(RE_{ND_val}, RE_{MD_val}, RE_{SD_val})| \quad (4.6)$$

subject to $w_1 + w_2 = 1$, with $w_1, w_2 \in [0, 1]$,

where w_1, w_2 are the weights for the objective functions, $obj_1(\cdot), obj_2(\cdot)$ are objective 1 and objective 2 functions, and $|\cdot|$ is to scale each objective function with min-max normalization (4.1). In this fitness score function, first, min-max normalization function is applied to transform the objective functions such that they have the same order of magnitude [54]. Then, a weighted sum between the two objectives is calculated to be the fitness score. This function is to be maximized.

By using a weighted sum for fitness score calculation, we introduce two weight parameters w_1 and w_2 , which correspond with obj_1 and obj_2 respectively. Varying the weights values effectively varies the importance of the two objective functions. For each pair of weights considered, the autoencoder model that yields the highest fitness score is selected as a candidate optimal model.

4.4.3 Autoencoder Feature Generation

The final step in the autoencoder degradation analysis component is autoencoder feature generation. As discussed, the latent space features and the reconstruction errors are generated using the autoencoder model. These two features are concatenated with non time series features, produced by the data preprocessing component, to form a new enriched dataset. This dataset is used for the final HVAC health condition classification. Because there are multiple candidate optimal autoencoder models, each candidate model is used to generate a new enriched dataset. Ultimately, all enriched datasets are passed to the health condition classification component for final HVAC health condition classification.

4.5 Health Condition Classification

The health condition classification component is the last component of the AE-ANN-HP framework. It is responsible for producing the final daily HVAC health conditions. In this component, an artificial neural network multi-class classification model is used to classify the final daily health condition of an HVAC system. Specifically, the classifier used for this component is a feed-forward neural network with an input layer, two hidden layers [55], and an output layer with three neurons for multi-class classification. The latent space features and the reconstruction errors produced by the autoencoder degradation analysis component and the non-time series features produced by the data preprocessing component are concatenated to form an enriched dataset for the classification task. This enriched dataset is used as the inputs to train the classifier.

As described in the previous section, for each pair of fitness score weights considered, the optimal autoencoder model, also referred to as the candidate model, is used to generate the latent space features and the reconstruction errors. This means that each pair of fitness score weights is associated with a distinctive dataset. Hence, a classification model is trained for each dataset. Additionally, hyperparameter optimization for each classification model is also conducted to ensure the best performance. The number of neurons in each hidden layer, the activation functions, and the learning rates are examples of hyperparameters that can be tuned for this classifier model. The objective function for the hyperparameter optimization of this multi-class classifier is the cross-entropy loss function. The classifier that yields the lowest cross-entropy loss value on the validation set across all tuned classifiers is selected as the final optimal classifier model.

4.6 Summary

In this chapter, we describe the proposed AutoEncoder and Artificial Neural Network based HVAC Health Prognostics (AE-ANN-HP) framework. This framework is a novel data-driven

predictive maintenance framework for HVAC systems without internal IoT sensors. The data preprocessing component transforms raw data into a better form that is suitable for use in machine learning. The autoencoder degradation analysis component analyses the relative degradation of each data sample. It produces a list candidate optimal autoencoder models and in turn generates a list of new enriched datasets. Lastly, the health condition classification component leverages a multi-class ANN classification model to classify the daily health conditions. Because multiple enriched datasets are generated for use in this component, an ANN classification model is trained with each enriched dataset. The model with the highest validation performance is deemed the optimal classifier model.

Chapter 5

Synthetic HVAC Data Generation with EnergyPlus

5.1 Introduction

In this chapter, we first describe a real dataset provided to us for this research and discuss the reasons this dataset is not used for this research. Then, we describe the synthetic data generation with EnergyPlus and show that the synthetic dataset resembles the features in the real counterpart such that the synthetic dataset is a valid alternative for the validation of AE-ANN-HP framework when the real dataset cannot be used.

5.2 Description of the Real Dataset

The dataset provided to us for this research, courtesy of our industry partner, contains HVAC power consumption and outside temperature data, both collected by a smart electric meter, from a medium-sized single-storey commercial building located in Toronto, Ontario, Canada. The meter readings are aggregated into 15-minute averages such that each hour contains 4 readings. Figure 5.1 shows the HVAC power consumption and outside temperature trends from the beginning of May to the end of October in 2022, spanning 6 months, and a zoomed in

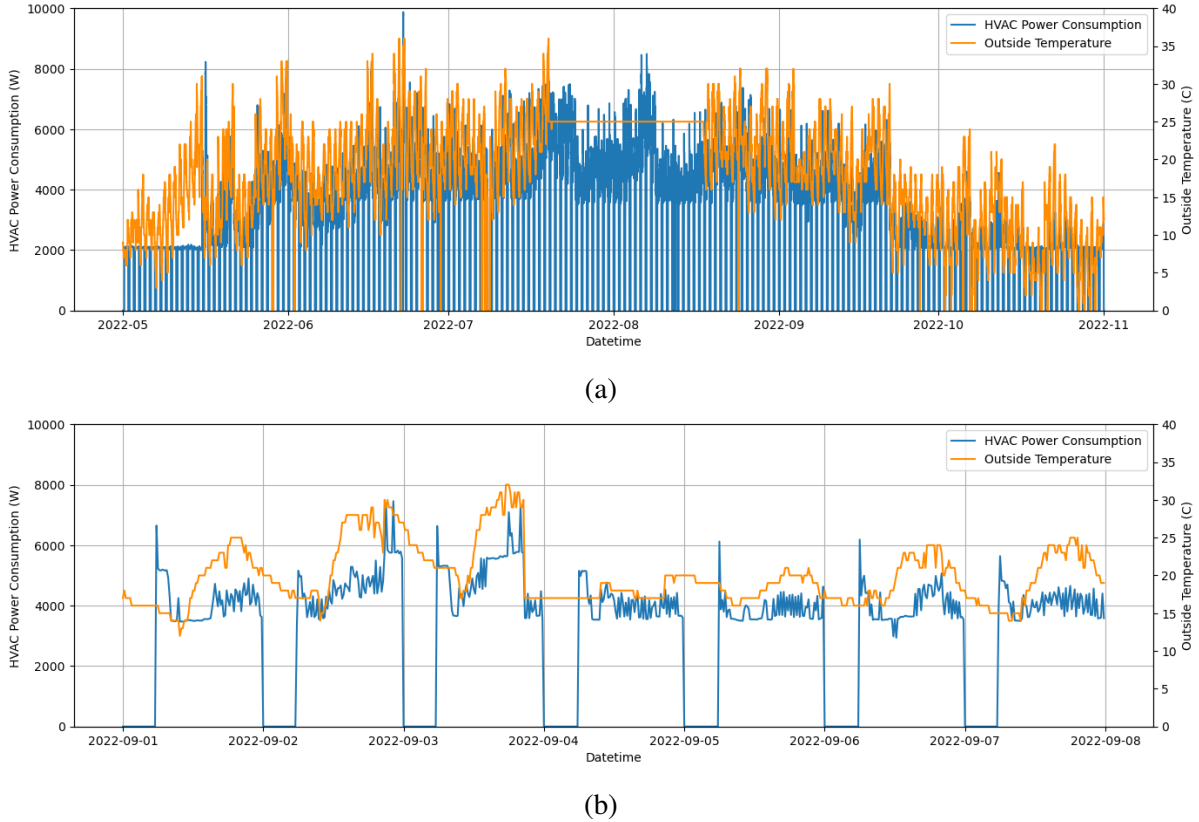


Figure 5.1: HVAC power consumption and outside temperature trends from the real dataset: (a) From May to October, (b) In the first week of September.

view of the trends for the first week of September. The outside temperature values remains at a constant 25°C from mid-July to mid-August because of a sensor malfunction. Several other instances of missing power consumption and outside temperature data are also present in the dataset. Nonetheless, the main reasons that the real dataset is not used for the training and testing of the AE-ANN-HP framework are as follows:

1. The maintenance records of the HVAC system are not easily accessible. Hence, it is not possible for us to label the dataset, i.e., determine the degradation status of the HVAC system for each sample in the dataset.
2. To generate data samples for the HVAC system operating in various degraded states and weather conditions for training the machine learning models, it would require extensive mechanical modifications to the HVAC system, which not only can interrupt the regular

service of the building, but also incur unnecessary labour and operational costs for the industry partner.

Despite this dataset being unsuitable for this research, it provided us with valuable information on the behaviors of the HVAC systems and enables a synthetic dataset to be generated for use in this research.

5.3 Synthetic Data Generation for HVAC system

For the simulation of the HVAC system’s power consumption, we adopt and modify the “5ZoneAir-Cooled” building template provided by EnergyPlus because of its popularity among researchers [56, 57]. The location of this building is set to Toronto, Ontario, Canada for consistency with the

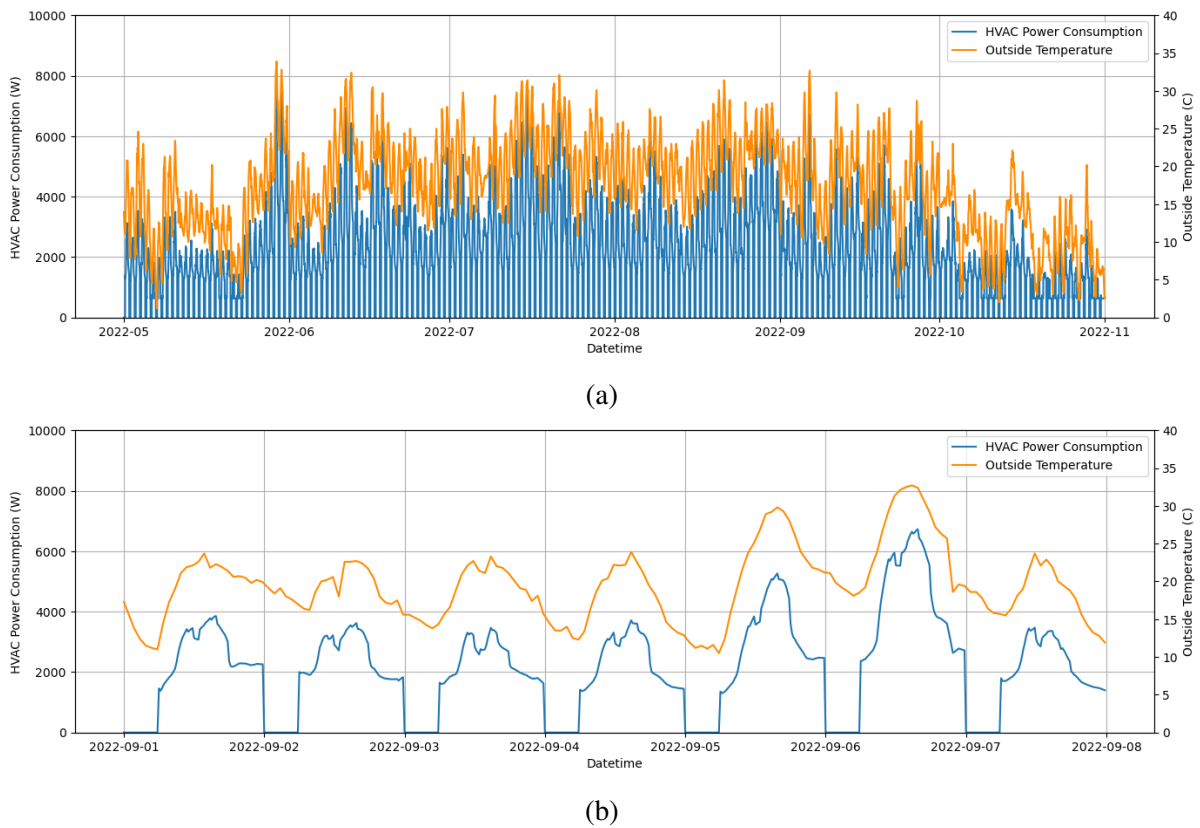
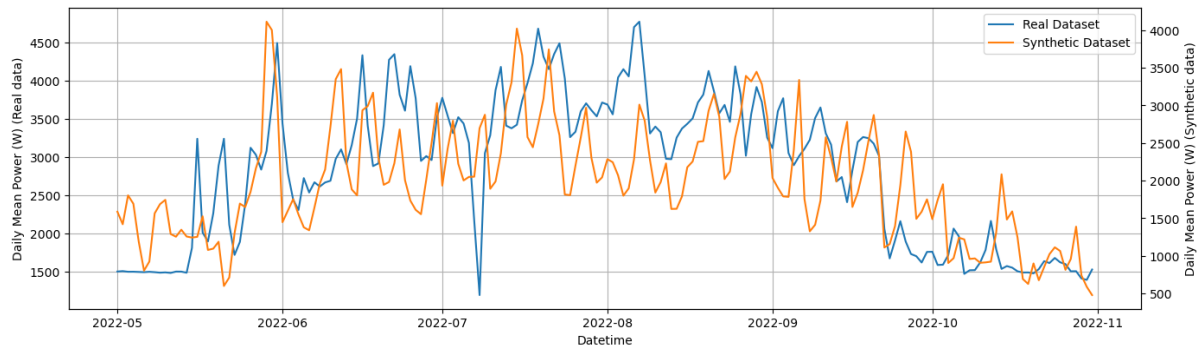


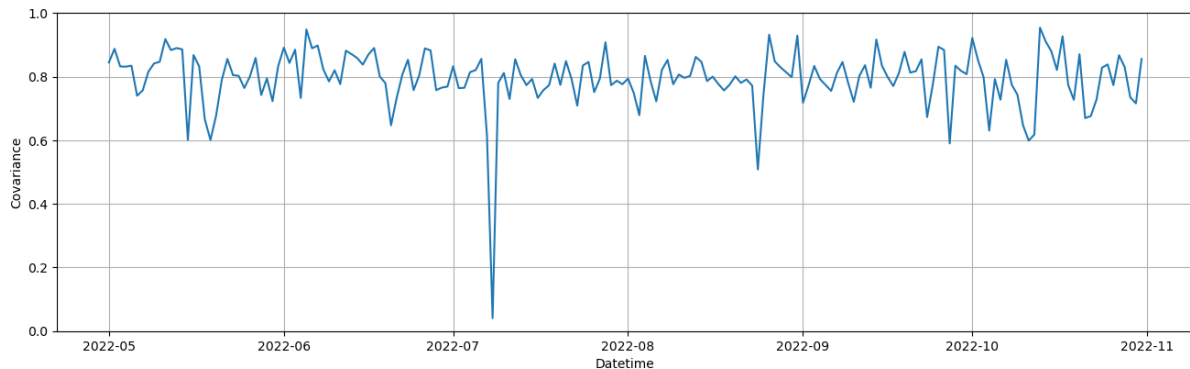
Figure 5.2: HVAC power consumption and outside temperature trends from a synthetically generated dataset using EnergyPlus: (a) From May to October, (b) In the first week of September.

real building location. To further the consistency, the weather information used for the simulations are the typical meteorological year (TMY) hourly weather datasets of the Pearson International Airport, located in close proximity to the building of interest. The TMY weather datasets are available on the website OneBuilding.Org [58]. Additionally, the outputs of EnergyPlus are also set to quarter-hourly averages. Lastly, the said building template and TMY weather dataset are fed into EnergyPlus to simulate HVAC behaviors. Figure 5.2 shows the synthetic HVAC power consumption and outside temperature trends generated with one of the TMY weather dataset and a zoomed in view of the trends in the first week of September.

To show that the synthetic dataset is similar to the real dataset, we compare the dataset using two metrics: daily mean of real and synthetic HVAC power consumption and the Pearson correlation coefficient of daily real and synthetic HVAC power consumption. The Pearson



(a)



(b)

Figure 5.3: Comparison between the real dataset and the synthetic dataset: (a) Daily mean HVAC power consumption, (b) Pearson correlation coefficient of daily HVAC power consumption

correlation coefficient r is given by

$$r(\mathbf{x}, \mathbf{y}) = \frac{\sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{y}_i - \bar{\mathbf{y}})}{\sqrt{\sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})^2 \cdot (\mathbf{y}_i - \bar{\mathbf{y}})^2}}, \quad (5.1)$$

where \mathbf{x} and \mathbf{y} are two vectors of the same length. A positive r means that the two vectors show similar trends, while a negative r means that the two vectors show opposite trends. As shown in Figure 5.3(a), although the daily mean power consumption vary because of the different weather conditions for the two datasets, both real and synthetic datasets show similar general upward then downward trend with peaks during the summer season. Additionally, figure 5.3(b) shows that the daily HVAC power consumption readings of the real and synthetic data have high Pearson correlation coefficients for all dates. Note that the dip in the graph corresponds to the abnormal daily HVAC power consumption trends resulted from sensor malfunctions. These two metrics show that the synthetic dataset exhibits similar features that are present in the real dataset. Hence, the synthetic dataset can be a valid alternative to the real counterpart.

5.4 Synthetic Data Generation for HVAC System with Degradation

To generate data for the moderate degradation (MD) and severe degradation (SD) conditions, we employ the EnergyPlus fault model for simulating HVAC systems in deteriorated health conditions. In this thesis, we simulate the power consumption of HVAC systems with various amount of dirtiness of the air filter for the variable volume supply fan, a subsystem of the HVAC system. Nassif [14] found that a moderately dirty air filter can increase the annual HVAC power consumption by an average of 14.3% while a very dirty air filter can increase the power consumption by 28.2%. Based on their findings, a fault intensity parameter in the fault model is adjusted such that the simulated annual HVAC power consumption exhibits an increase of 14.3% for moderate degradation condition from normal and an increase of 28.2%

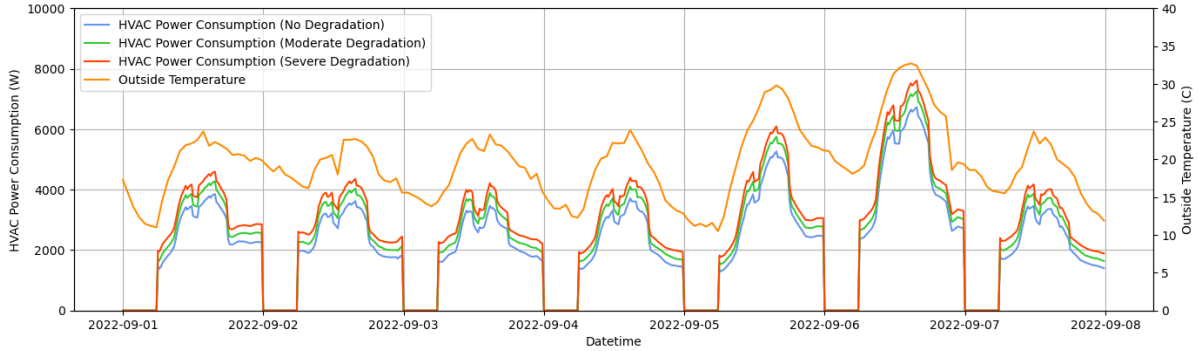


Figure 5.4: Comparison of the HVAC power consumption in no degradation, moderate degradation, and severe degradation conditions.

for severe degradation condition from normal. Figure 5.4 illustrates the effects of the degradation simulations. In moderate degradation and severe degradation conditions, the HVAC power consumption show slight increase over no degradation condition. However, because of the deterministic nature of the EnergyPlus simulation, i.e., the same input produces the same output every time the simulation is ran, the general shapes of the daily trend for all three conditions remain the same.

5.5 Summary

This chapter covers the synthetic data generation process with EnergyPlus. The reason that a synthetic dataset is needed is that while the real data provided by our industry partner provides valuable information, the dataset do not contain enough data to support the training and testing of the framework. Next, we show that the simulated HVAC data is a valid alternative because it shares features that are present in the real dataset. Finally, we describe the simulation of HVAC system with varying levels of degradation using EnergyPlus fault model.

Chapter 6

AE-ANN-HP Framework Evaluation

6.1 Introduction

In this chapter, we describe the evaluation of the AE-ANN-HP framework. First, we describe the dataset and the evaluation metrics used for the evaluation. Next, we conduct four experiments and discuss the results. The first experiment establishes the baseline performance using existing health prognostics methods. The next three experiments apply three different autoencoders to the AE-ANN-HP framework. Finally, we compare the performances and discuss the findings.

6.2 Dataset Split

As discussed in Chapter 5, the evaluation of the AE-ANN-HP framework is done on a synthetic dataset using the EnergyPlus simulation software. Specifically, 6 typical meteorological year (TMY) hourly weather datasets are used to simulate the HVAC power consumption (6 rows in Figure 6.1). For each weather dataset, the HVAC power consumption for no degradation, moderate degradation, and severe degradation conditions are simulated (3 columns in Figure 6.1). As previously mentioned, the data resolution is set to quarter-hour, thus each day contains 96 readings of the two features. The HVAC simulations are ran for the period from

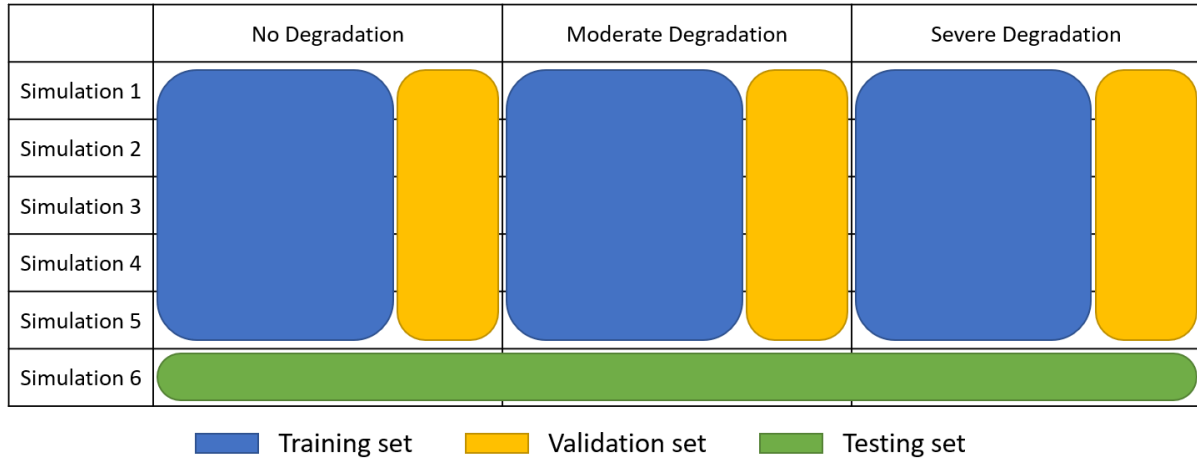


Figure 6.1: Training, validation, and testing split of the dataset.

May 1 to October 31, which contains 184 days. Overall, the final combined dataset contains HVAC data for 3,312 days or 317,952 individual readings. Data generated by all 6 simulations in no degradation condition are assigned with the no degradation label. The same is true for data generated in the two degraded conditions. Hence, this dataset is a balanced dataset as each health condition contains the same number of simulated days.

The dataset is then split into training, validation, and testing set. Figure 6.1 illustrates this split. First, the synthetic data generated by one of the TMY hourly weather datasets (Simulation 6 in Figure 6.1) is set aside and reserved for testing the models' generalization performance. Then, for the remaining 5 simulations, for each health condition, 20% of the data is split into validation set and the remaining 80% of the data is used for training. Dedicating an entire simulation for testing is a better strategy than simply splitting another 20% of all data, because an entire simulation contains HVAC data for each day from May to October for each health condition, which covers a wider range of weather conditions and HVAC power consumption trends. In result, there are 2,208 days of data in the training set, 552 days in the validation set, and another 552 in the testing set.

6.3 Evaluation Metrics

To evaluate the model performance, we use confusion matrix, multi-class F1 score with macro averaging, and the one-vs-rest (OVR) area under receiving operating characteristics curve (AU-ROC) with macro averaging.

Confusion matrix is commonly used for the evaluation of multi-class classification tasks because it shows the model's ability to predict each class as well as the classes it misclassifies as when it fails to predict the actual class. Figure 6.2 represents a multi-class confusion matrix with true positive (TP), true negative (TN), false positive (FP), and false negative (FN) labeled with respect to the class C_1 . TP represents the number of correctly classified positive samples. TN represents the number of correctly classified negative samples. FP represents the number of actual negative samples that are misclassified as positive samples. FN represents the number of actual positive samples that are misclassified as negative samples. Additionally, from the confusion matrix, precision and recall can be calculated by (6.1) and (6.2) respectively. Precision represents the proportion of predicted positive samples which are correctly classified. Recall represents the proportion of actual positive samples which are correctly classified as positive.

$$Precision = \frac{TP}{TP + FP} \quad (6.1)$$

$$Recall = \frac{TP}{TP + FN} \quad (6.2)$$

		Predicted Class		
		C_1	C_2	C_3
Actual Class	C_1	TP	FN	
	C_2	FP	TN	
	C_3			

Figure 6.2: A multi-class confusion matrix. TP, TN, FP, FN are labeled in class C_1 's perspective.

F1 score is a popular accuracy measure for machine learning models. For binary classification, it is defined as the harmonic mean of precision and recall. The equation to calculate F1 score is given by (6.3). A higher F1 score indicates a higher model performance. A F1 score of 1 represents a model that can correctly classify all instances. In this thesis, we use the macro-averaging method to calculate the F1 score in multi-class setting because the testing dataset is a balanced dataset, i.e., it contains the same number of samples for each class. The macro averaged F1 score is the unweighted average of the F1 score for each individual class. It is given by (6.4).

$$F_1 = 2 \frac{\textit{Precision} \cdot \textit{Recall}}{\textit{Precision} + \textit{Recall}} \quad (6.3)$$

$$F_1^{\textit{macro}} = \frac{1}{n} \sum_{i=1}^n F_1^{C_i} \quad (6.4)$$

$$FPR = \frac{FP}{FP + TN} \quad (6.5)$$

Receiver operating characteristic curve (ROC) provides a visual representation of a binary classifier's performance with varying decision thresholds. In an ROC curve, the true positive rates (TPR) evaluated at varying thresholds is plotted against the false positive rates (FPR). TPR is the same as recall, which is given by (6.2). FPR is the proportion of actual

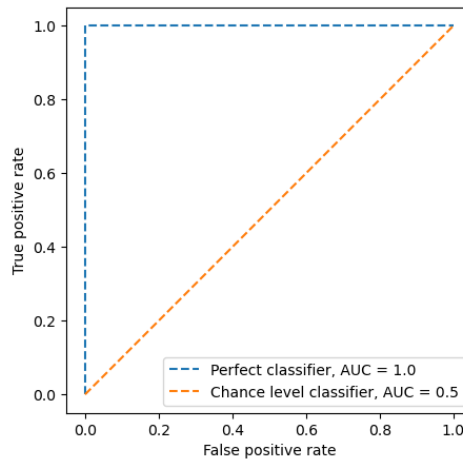


Figure 6.3: ROC curves for a perfect classifier and a chance level classifier.

negative samples which are misclassified as positive, calculated by (6.5). The ROC curve has both a domain and a range of $[0, 1]$. For a model that classifies no better than random chance, the ROC curve resembles a diagonal line from $(0, 0)$ to $(1, 1)$. On the other hand, for a perfect classifier, the ROC curve follows the y-axis from $(0, 0)$ to $(0, 1)$ then a horizontal line from $(0, 1)$ to $(1, 1)$.

Area under ROC curve (AUROC) measures the enclosed space between the receiver operating characteristic (ROC) curve and the x-axis from 0 to 1. This metric measures the probability for a model to classify a random positive instance as more positive than a random negative instance. While the AUROC score can be arbitrarily lower than 0.5, a chance level classifier would score an AUROC of 0.5 and a perfect classifier would score 1.0, as shown in Figure 6.3. AUROC is useful for evaluating the performance of the AE-ANN-HP framework because the three health conditions have implied ranking, i.e., severe degradation is more positive than moderate degradation, which in turn is more positive than no degradation.

6.4 Experiments & Results

In this section, we present four experiments to assess the performance of the AE-ANN-HP framework. In the first experiment, we evaluate existing methods for HVAC health prognostics on the same dataset to establish a performance baseline for the AE-ANN-HP framework to compare with. We consider three methods: artificial neural networks (ANN) [9], support vector machine (SVM) [9], and decision tree (DT) [12]. In the subsequent three experiments, we evaluate AE-ANN-HP with the use of three different autoencoders, namely dense autoencoder (DAE), convolutional autoencoder (CAE), and long short-term memory autoencoder (LSTMAE). In experiment II with DAE, the objective is to study the effect of the inclusion of a simple autoencoder over directly using a classifier for HVAC health condition classification. In experiment III and IV with CAE and LSTMAE, the objective is to study the performance gain when using autoencoders with more complex architectures.

All machine learning algorithms are trained with three different seeds and the performances are averaged to reduce the chance of luck. For experiment I with three baseline algorithms, the results are averaged over three trials. On the other hand, for experiment II, III, and IV with AE-ANN-HP framework, because two machine learning algorithms are involved, the results are generated as follows: all autoencoders are first trained with three different seeds, the optimal hyperparameters for the autoencoder models are selected based on the average fitness score of three trials. Then, all three trained autoencoders with the optimal hyperparameters are used for the generation of enriched datasets. Lastly, for each enriched dataset, a classifier is trained with three different seeds. Hence, the final results of the experiments with the AE-ANN-HP framework are averaged over nine trials.

6.4.1 Experimental Setup

All four experiments are performed on a desktop PC with an Intel Core i7-9700k CPU, an NVIDIA GeForce RTX 2080 GPU, and 32 GB DDR4 memory. In experiment I, the baseline ANN model is implemented in Python programming language using Tensorflow library with Keras API. The baseline SVM and baseline DT models are implemented in Python using Scikit Learn library. In experiment II - IV, the DAE, CAE, LSTMAE, and ANN classification models are implemented in Python using Tensorflow library with Keras API. The training of LSTMAE models are accelerated with GPU using cuDNN library. Pandas and Numpy libraries are used for dataset manipulation. The results figures are generated using Matplotlib library. The three seeds used for training the machine learning algorithms are 1, 2, 3.

6.4.2 Experiment I: Performance Baseline

In the first experiment, we evaluate existing HVAC health prognostics methods on the aforementioned synthetic HVAC dataset. The results from the existing methods are used for comparison with the AE-ANN-HP framework. We consider three machine learning models: ANN [9], SVM [9], and DT [12].

Experiment Process

To ensure fairness across all experiments, the same data are used for training all three baseline models as well as three AE-ANN-HP models. The dataset is prepared using the data preprocessing component in the AE-ANN-HP framework. Recall that the electric meter readings have a 15-minute resolution, thus each day contains a total of 96 readings. The daily HVAC power consumption and outside temperature readings from midnight to 6 A.M. (24 readings) are removed because of inactivity, which reduces the number of readings from 96 to 72 per day. Statistical features including daily minimum, daily maximum, daily average, and daily standard deviation are calculated for both time series features. Time derived features including month of year, day of year, and day of week are extracted and encoded with trigonometric encoding (4.2). After data preprocessing, the dataset contains a total of 158 features, including 2 time-series features each with 72 time steps, 8 statistical features, and 6 date derived features. For each baseline model, hyperparameter tuning is also conducted to achieve the best model performance.

The baseline ANN model is the same as AE-ANN-HP’s final classification model. The difference is that for the experiment with baseline ANN model, after preprocessing, the features are passed directly to the ANN model for classification, skipping the autoencoder degradation analysis step. The network is set to contain two hidden layers [55]. Adam is used as the optimizer. To control model overfitting, the validation set is used to monitor the validation

Table 6.1: List of hyperparameters and their search space for tuning the baseline ANN model.

Hyperparameter	Description	Search space
hidden_units	Number of neurons in each hidden layer.	Input size \times {2, 3, 4} ¹
activation	The activation function for each layer except for last layer (SoftMax).	{relu, sigmoid, tanh}
lr	Learning rate for Gradient Descent.	{0.01, 0.001, 0.0001}

¹Here the number of neurons in each hidden layer of the ANN depends on the size of the input layer. For example, the baseline ANN model has an input size of 158. The search space is therefore {316, 474, 632}. For a classifier in AE-ANN-HP framework with 42 inputs, the search space is then {84, 126, 168}.

cross-entropy loss for early stopping of the training process. The minimum delta for validation loss is set to 0.0001 and the early stopping patience is set to 100 epochs. The hyperparameters used for tuning the ANN model are listed in Table 6.1. The model with the lowest validation cross-entropy loss is used for reporting testing performance.

Similar to the baseline ANN model, the baseline SVM and DT models are also trained on the preprocessed dataset containing 158 features. Unlike the baseline ANN model, the baseline SVM and DT models are trained until convergence. Hence, the validation set is only used for hyperparameter tuning. The hyperparameters used for tuning the two models are listed in Table 6.2 and Table 6.3. For the baseline SVM model, the solver is restricted to 1 million iterations to mitigate non-convergence cases. Because the SVM and DT models do not require the validation set for early stopping of the training process, after the tuning process, these models are re-trained with their best hyperparameters with the combined training and validation set. The re-trained models are used to report testing performance.

Table 6.2: List of hyperparameters and their search space for tuning the baseline SVM model.

Hyperparameter	Description	Search space
C	l_2 regularization parameter.	50 exponentially spaced values with range [1, 1000]
kernel	The kernel function.	{linear, poly, rbf, sigmoid}

Table 6.3: List of hyperparameters and their search space for tuning the baseline DT model.

Hyperparameter	Description	Search space
criterion	Function that measures the quality of splits.	{gini, entropy}
max_depth	The maximum depth of the tree.	{10, 20, 30, 40, 50, inf}
min_samples_split	The minimum number of samples needed to split an internal node.	10 linearly spaced values with range [2, 11]
min_samples_leaf	The minimum number of samples needed to be at a leaf node.	10 linearly spaced values with range [1, 10]
max_features	The number of features to consider for finding the best split.	{sqrt, log2}

Results Analysis

Recall that we attempt to determine the health condition of HVAC systems using only daily HVAC power consumption and outside temperature data. The objective of this experiment is to investigate the performance of existing methods in a scenario where the number of features are limited. Figure 6.4 shows the confusion matrices against the testing set for all three baseline models. Note that the values in the confusion matrices are averaged over three trials. It is evident from the confusion matrices that the ANN and SVM models are somewhat capable of accurately classifying the daily HVAC health conditions. However, upon further inspection, a common trend can be noticed that the models misclassify many moderate degradation (MD) instances as severe degradation (SD) and many no degradation (ND) instances as moderate degradation (MD), but rarely misclassify moderate degradation (MD) instances as no degradation (ND) and severe degradation (SD) instances as moderate degradation (MD). This suggests that they tend to misclassify some data samples to be in a more degraded condition than their actual conditions. On the other hand, the DT model shows inferior performance than the other

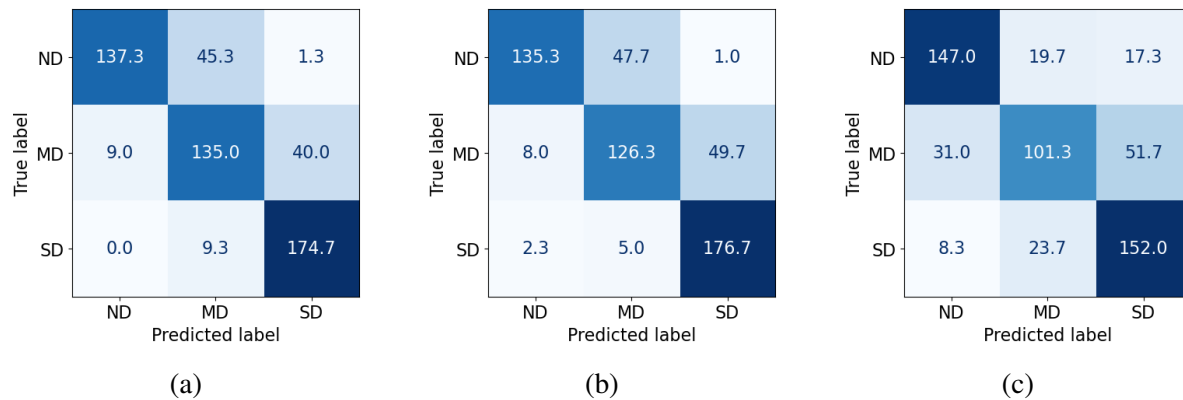


Figure 6.4: Comparison of confusion matrices of baseline models against the testing set: (a) ANN, (b) SVM, (c) DT. All values are averaged over three trials.

	ANN	SVM	DT
F1	0.8091	0.8005	0.7654
AUROC	0.9534	0.9422	0.9111

Table 6.4: Comparison of F1 and AUROC scores of baseline models against the testing set. All scores are averaged over three trials.

two baseline models and struggles with classifying data samples in moderate degradation class. The simultaneous high AUROC scores and low F1 scores exhibited by the three baseline models show that Despite that the models' likelihood of ranking a random data sample in a more degraded class as more degraded than a random data sample in a less degraded class, they are not as good at achieving a balance between precision and recall. Consequently, the best performing model, the baseline ANN model, yields an macro-averaged AUROC of 0.9534 but a macro-averaged F1 score of 0.8091.

The AE-ANN-HP framework aims to improve the health condition classification performance by introducing the autoencoder degradation analysis component to the process. We experiment with DAE, CAE, and LSTMAE in the subsequent experiments.

6.4.3 Experiment II: AE-ANN-HP with Dense Autoencoder

The AE-ANN-HP framework adds an additional autoencoder component to extract features from the daily HVAC data before classifying the health condition of the system. In this experiment, we evaluate AE-ANN-HP with a dense autoencoder (DAE) for the autoencoder degradation analysis component. We refer to this variant of the framework as DAE-ANN-HP for the rest of the chapter. The goal of this experiment is to study the effect of using a simple autoencoder for feature generation.

Experiment Process

The data preprocessing is the same as described in experiment I. The preprocessed data contains 158 features including 144 time series features, 8 statistical features, and 6 date derived features. The 144 time series features are used to train the DAE model. This model is trained on no degradation training set. MSE (2.2) is used as the loss function. The no degradation validation set is used to monitor the validation loss for early stopping of the training process. Table 6.5 lists the hyperparameters considered for tuning the DAE model. As discussed in Chapter 4, the candidate optimal models are selected by calculating their fitness scores (4.6).

Table 6.5: List of hyperparameters and their search space for tuning the DAE model.

Hyperparameter	Description	Search space
encoder_topology	The number of neurons in each hidden layer of the encoder.	1 hidden layer: {72, 54, 36, 27} 2 hidden layers: {(72, 54), (72, 36), (72, 27), (54, 36), (54, 27), (36, 27)}
activation	The activation function for each layer.	{relu, sigmoid, tanh}
min_loss_delta	The minimum decrease in validation loss for early stop of training process.	{0.01, 0.001, 0.0001}
lr	Learning rate for gradient descent.	{0.01, 0.001, 0.0001}
dropout	The dropout rate of each layer during training.	{0, 0.1, 0.2}

We consider 5 pairs of values for weight parameters w_1 and w_2 for fitness score calculation: $\{(0, 1), (0.25, 0.75), (0.5, 0.5), (0.75, 0.25), (1, 0)\}$. Recall that conceptually, w_1 corresponds to the importance of obj_1 , which assesses the autoencoder’s ability to minimize the reconstruction errors for no degradation samples. w_2 corresponds to obj_2 , which assesses the autoencoder’s ability to differentiate the reconstruction errors for all three health conditions. Varying the weights varies the relative importance of the two objective functions. For example, the weights pair $w_1 = 0$ and $w_2 = 1$ means that only obj_2 is concerned, while $w_1 = 0.5$ and $w_2 = 0.5$ means that the two objectives contribute equally to the fitness score calculation. After all DAE models are trained, a fitness score is calculated using the whole validation set for each model for each pair of weights w_1 and w_2 . Because we consider 5 weights pairs, 5 candidate DAE models are selected to generate 5 new enriched datasets for final HVAC health condition classification.

To generate a new enriched dataset, the latent space features and the reconstruction errors are generated for all data samples in the training, validation, and testing set. The generated features are then concatenated with the statistical features and date derived features to form the new training, validation, and testing set for final classification. The health condition classifier in AE-ANN-HP has the same structure and hyperparameters considered for tuning as the baseline ANN model in Experiment I. One classifier is trained and hyperparameter tuned for each new enriched dataset. As explained previously, each classifier is also trained with three different

seeds and the results are averaged. The classifier with the best validation cross-entropy loss is used to report testing performance.

Results Analysis

The best DAE models, i.e., the ones with the highest fitness score, for each pair of fitness score weights parameter w_1 and w_2 are listed in Table 6.6. Note that as explained earlier, because three different seeds are used during training, each candidate represents a collection of three trained autoencoders. Also note that candidate 1 and 2 have the same hyperparameters, thus candidate 2 is ignored. A total of 4 candidates are then used to generate enriched datasets for the training and tuning of the health condition classifiers. Table 6.7 shows the average validation cross-entropy loss of the classifiers trained on the datasets associated with the candidate DAE models. The best performing classifier is the one that is trained with the dataset produced by DAE candidate 1, with an average validation cross-entropy loss of 0.3013 across nine trials. It is also evident that as w_1 increases and w_2 decreases, the average cross-entropy loss increases. This suggests that objective 2 in the fitness function, which measures the autoencoder’s ability to differentiate between data samples in each class, plays a more important role than objective 1 in the framework’s final health condition classification performance. This classifier is used to

Table 6.6: Hyperparameters of the five candidate DAE models.

candidate	w_1	w_2	topology	activation	min_loss_delta	lr	dropout
1	0	1	(54, 27)	relu	0.01	0.0001	0.0
2	0.25	0.75	(54, 27)	relu	0.01	0.0001	0.0
3	0.5	0.5	(36, 27)	sigmoid	0.01	0.01	0.1
4	0.75	0.25	(36, 27)	sigmoid	0.001	0.0001	0.0
5	1	0	(72, 54)	tanh	0.0001	0.001	0.0

Table 6.7: Final health condition classification validation losses of DAE-ANN-HP.

Candidate	Validation cross-entropy loss
1	0.3013
3	0.3260
4	0.3479
5	0.3854

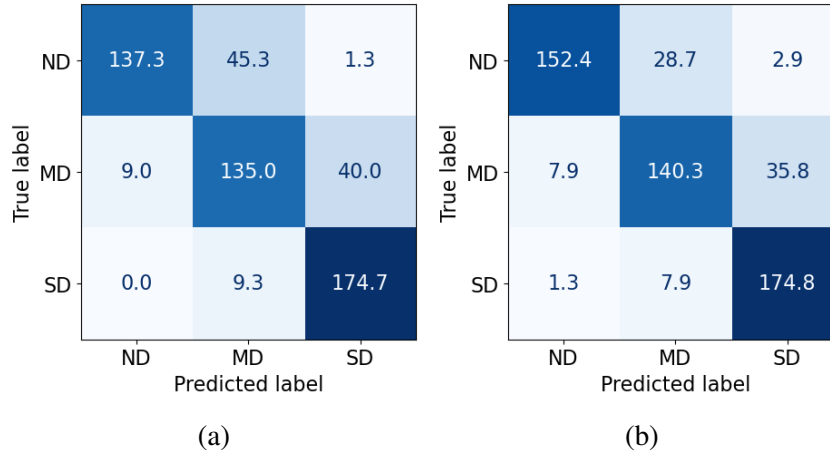


Figure 6.5: Comparison of confusion matrices against the testing set: (a) baseline ANN, (b) DAE-ANN-HP. Confusion matrix for baseline ANN is averaged over three trials. Confusion matrix for DAE-ANN-HP is averaged over nine trials.

Table 6.8: Comparison of F1 and AUROC scores between DAE-ANN-HP and baseline ANN against the testing set. Scores for baseline ANN are averaged over three trials. Scores for DAE-ANN-HP are averaged over nine trials.

	Baseline ANN	DAE-ANN-HP
F1	0.8091	0.8463 (+4.60%)
AUROC	0.9534	0.9664 (+1.36%)

report testing performance of DAE-ANN-HP. The confusion matrix of DAE-ANN-HP against the testing set is compared to that of baseline ANN in Figure 6.5.

The confusion matrix suggests DAE-ANN-HP exhibits similar problem to the baseline models in which the classifier has the tendency to classify health conditions as more deteriorated than actual. Nonetheless, it correctly classifies significantly more no degradation (ND) and moderate degradation (MD) instances. DAE-ANN-HP also shows a noticeable improvement in F1 score and AUROC score over the baseline ANN model, as suggested by Table 6.8.

6.4.4 Experiment III: AE-ANN-HP with Convolutional Autoencoder

In this experiment, we evaluate AE-ANN-HP with a convolutional autoencoder (CAE) for the autoencoder degradation analysis component. The goal of this experiment is to study how

much the classification performance can be improved by using a more complex autoencoder over a simple DAE. A convolutional autoencoder is also specialized in extracting temporal relationships within time series features. We refer to this variant of the framework as CAE-ANN-HP for the rest of this chapter.

Experiment Process

Similar to the previous experiment, the daily HVAC power consumption and outside temperature readings are used to train the CAE model. Because CAE utilises convolution operations, the input data are reshaped to (number of time steps, number of features), i.e. $(72, 2)$, to be fed into the CAE model. The CAE model used for this experiment is structured to have three pairs of 1D convolution and maxpooling layers for the encoder. Each convolution layer contains 16 filters for feature extraction. The size of the filters is a hyperparameter to be tuned. The stride is set to 1. Zeros are padded to the output of convolution operations to retain its original dimension. The maxpooling layers are set to reduce the input dimension by a factor of 2. The output of the last maxpooling layer is flattened and transformed into the latent space with an additional dense layer. The number of neurons in the dense layer is another hyperparameter to be tuned. In the decoder part, three pairs of upsampling and 1D deconvolution layers are used. The hyperparameters are the same as those in the convolution and maxpooling layers. An additional 1D convolution layer with 2 filters transforms the final output back to $(72, 2)$. Table 6.9 details the hyperparameters tuned for the CAE model. Dropout is not tuned for this

Table 6.9: List of hyperparameters and their search space for tuning the CAE model.

Hyperparameter	Description	Search space
latent_size	The size of the latent space of CAE.	{72, 54, 36, 27}
kernel_size	The size of the kernels applied in each convolution and deconvolution layer.	{3, 5, 7, 9}
activation	The activation function for each layer.	{relu, sigmoid, tanh}
min_loss_delta	The minimum decrease in validation loss for early stop of training process.	{0.01, 0.001, 0.0001}
lr	Learning rate for gradient descent.	{0.01, 0.001, 0.0001}

model because convolutional neural networks are unlikely to suffer from overfitting for models with a relatively low number of trainable parameters [59]. The rest of the experiment process including the selection of candidate CAE models and the training of health condition classifier is the same as in the previous experiment.

Table 6.10: Hyperparameters of the five candidate CAE models.

candidate	w_1	w_2	latent	kernel	activation	min_loss_delta	lr
1	0	1	54	9	relu	0.001	0.01
2	0.25	0.75	54	9	relu	0.001	0.01
3	0.5	0.5	72	5	tanh	0.01	0.01
4	0.75	0.25	54	7	tanh	0.0001	0.01
5	1	0	36	3	tanh	0.0001	0.01

Table 6.11: Final health condition classification validation losses of CAE-ANN-HP.

candidate	validation cross-entropy loss
1	0.3044
3	0.3684
4	0.4022
5	0.3904

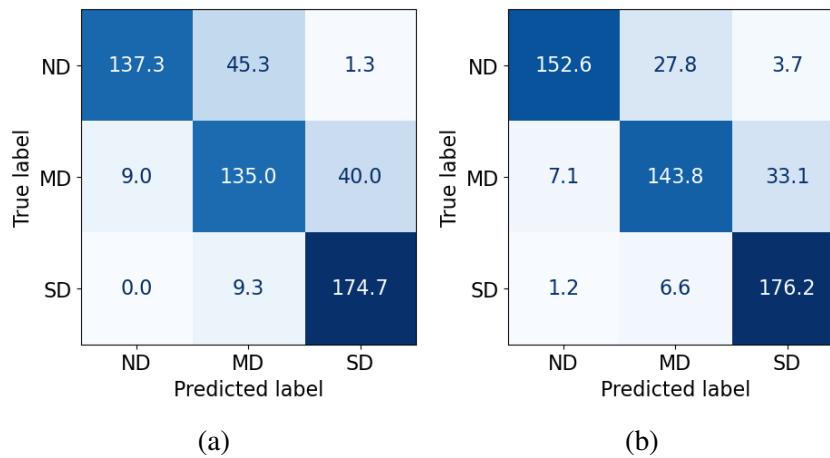


Figure 6.6: Comparison of confusion matrices against the testing set: (a) baseline ANN, (b) CAE-ANN-HP. Confusion matrix for baseline ANN is averaged over three trials. Confusion matrix for CAE-ANN-HP is averaged over nine trials.

Table 6.12: Comparison of F1 and AUROC scores between CAE-ANN-HP and baseline ANN against the testing set. Scores for baseline ANN are averaged over three trials. Scores for CAE-ANN-HP are averaged over nine trials.

	Baseline ANN	CAE-ANN-HP
F1	0.8091	0.8555 (+5.73%)
AUROC	0.9534	0.9746 (+2.22%)

Results Analysis

The best CAE model for each pair of fitness score weights parameter w_1 and w_2 is listed in Table 6.10. Similar to the DAE-ANN-HP experiment, candidate 1 and 2 have the same hyper-parameters, thus candidate 2 is ignored. Hence, 4 candidates are used to generate the enriched datasets for training and tuning the health condition classifier. Table 6.11 shows the average validation cross-entropy loss over nine trained classifiers against enriched datasets generated by each candidate autoencoder model. Candidate 1 produces the lowest validation cross-entropy loss of 0.3044. Figure 6.6 shows the comparison of confusion matrices between CAE-ANN-HP and the baseline ANN against the testing set. As shown in Table 6.12, CAE-ANN-HP scores a F1 score of 0.8555 and an AUROC score of 0.9746, which represent 5.73% and 2.22% improvement over the baseline ANN model. The performances of CAE-ANN-HP is also slightly better than DAE-ANN-HP in terms of average F1 and AUROC scores. The higher performance of CAE-ANN-HP suggests that an autoencoder with a more complex structure is better suited for the autoencoder degradation analysis component of the AE-ANN-HP framework.

6.4.5 Experiment IV: AE-ANN-HP with Long Short-Term Memory Autoencoder

In the last experiment, we evaluate AE-ANN-HP with a long short-term memory autoencoder (LSTMAE) for the autoencoder degradation analysis component. This experiment is to study how well the framework works with an autoencoder that is specialized to work with time series data. We refer to this variant as LSTMAE-ANN-HP for the rest of this chapter.

Table 6.13: List of hyperparameters and their search space for tuning the LSTMAE model.

Hyperparameter	Description	Search space
encoder_topology	The number of LSTM cells in each LSTM layer of the encoder.	1 LSTM layer: {72, 54, 36, 27} 2 LSTM layers: {(72, 54), (72, 36), (72, 27), (54, 36), (54, 27), (36, 27)}
min_loss_delta	The minimum decrease in validation loss for early stop of training process.	{0.01, 0.001, 0.0001}
lr	Learning rate for gradient descent.	{0.01, 0.001, 0.0001}
dropout	The dropout rate of each layer during training.	{0, 0.1, 0.2}

Experiment Process

Similar to CAE in the previous experiment, LSTMAE also requires an input shape of (number of time steps, number of features). Hence, the inputs are again reshaped to (72, 2) to be fed into the LSTMAE model. To tune the number of LSTM layers in the encoder and the decoder of the LSTMAE model, modifications to the LSTM layer mappings need to be made. For models with one LSTM layer in the encoder and the decoder, the model structure discussed in Section 2.3.2 is used. On the other hand, for models with two LSTM layers, all layers in the encoder except for the last layer are configured to have many-to-many mapping, i.e., they return the full output sequences from all 72 time steps, which are fed into the subsequent LSTM layer, whereas the last layer has a many-to-one mapping, i.e., only the output from the last time step is returned. All layers in the decoder return the full output sequences. In order to use the cuDNN library to accelerate LSTMAE model training with GPU, tanh activation has to be used for all LSTM layers in the model. Table 6.13 shows the search spaces for the hyperparameters considered for tuning the LSTMAE model. The rest of the experiment process including the selection of candidate LSTMAE models and the training of health condition classifiers is the same as in the previous experiment.

Table 6.14: Hyperparameters of the five candidate LSTMAE models.

candidate	w_1	w_2	encoder_topology	min_loss_delta	lr	dropout
1	0	1	(72, 54)	0.01	0.01	0.2
2	0.25	0.75	(72, 54)	0.01	0.01	0.2
3	0.5	0.5	(72, 54)	0.01	0.01	0.2
4	0.75	0.25	(72, 54)	0.001	0.01	0.0
5	1	0	72	0.0001	0.01	0.0

Table 6.15: Final health condition classification validation losses of LSTMAE-ANN-HP.

candidate	validation cross-entropy loss
1	0.3036
4	0.3663
5	0.4049

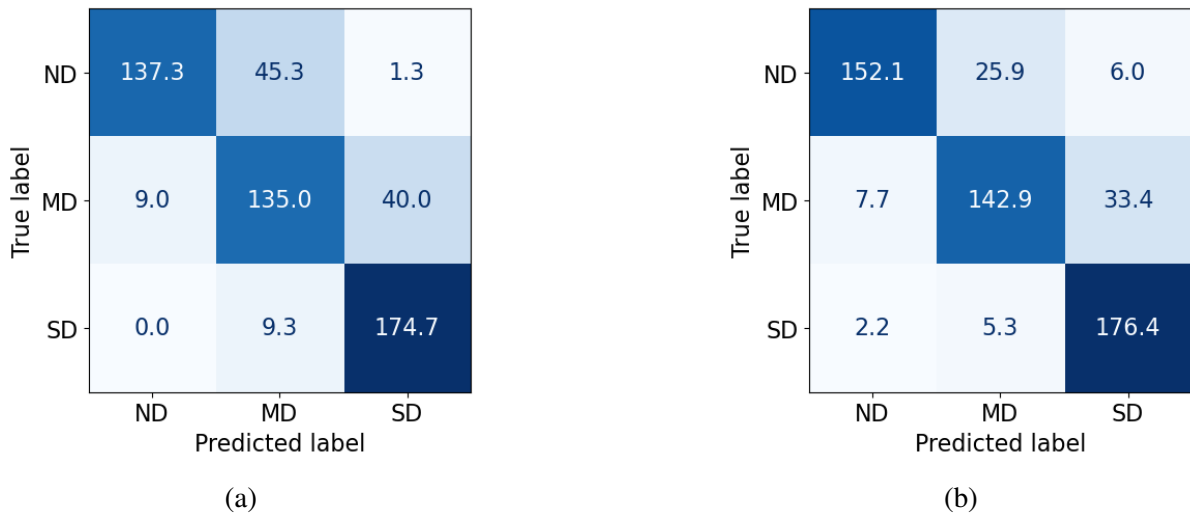


Figure 6.7: Comparison of confusion matrices against the testing set: (a) baseline ANN, (b) LSTMAE-ANN-HP. Confusion matrix for baseline ANN is averaged over three trials. Confusion matrix for LSTMAE-ANN-HP is averaged over nine trials.

Table 6.16: Comparison of F1 and AUROC scores between LSTMAE-ANN-HP and baseline ANN against the testing set. Scores for baseline ANN are averaged over three trials. Scores for DAE-ANN-HP are averaged over nine trials.

	Baseline ANN	LSTMAE-ANN-HP
F1	0.8091	0.8533 (+5.46%)
AUROC	0.9534	0.9725 (+2.00%)

Results Analysis

The LSTMMAE model with the highest fitness score for each pair of fitness score weights parameter w_1 and w_2 is listed in Table 6.14. Note that for LSTMMAE-ANN-HP, candidates 1, 2, and 3 all share the same hyperparameters. Consequently, only 3 candidates are used to generate the enriched datasets for training and tuning the health condition classifier. Table 6.15 shows the average validation cross-entropy loss over nine trials for each candidate autoencoder model. The classifier trained on the enriched datasets generated by LSTMMAE candidate 1 produced the lowest average validation cross-entropy loss of 0.3036. Figure 6.7 shows the comparison of confusion matrices between LSTMMAE-ANN-HP and the baseline ANN model against the testing set. Similar to the previous two experiments with AE-ANN-HP framework, LSTMMAE-ANN-HP is significantly better at classifying no degradation (ND) and moderate degradation (MD) instances than the baseline ANN model. As shown in Table 6.16, LSTMMAE-ANN-HP yields an average F1 score of 0.8533 and an average AUROC score of 0.9725. This performance is slightly better than that of DAE-ANN-HP and very close to that of CAE-ANN-HP.

6.5 Discussion

To evaluate the performance of AE-ANN-HP framework, we test the framework with three types of autoencoders, which are DAE, CAE, and LSTMMAE. We also evaluate three existing methods, which include ANN, SVM, and DT, on the same dataset for performance comparison. The F1 scores and AUROC scores for all 6 experimented methods are compared in Figure 6.8. For each method, the scores for all trials are plotted with box plot. The colored boxes in the two figures represent the inter quantile range (IQR) and the small white square in each box represent the average value. It is evident that all three experiments with the proposed framework yielded better performances comparing to the baseline models. When comparing the performances within the three variations of the framework, although the use of a dense autoencoder could theoretically produce the best performance, the results for 9 trials show high inconsistency,

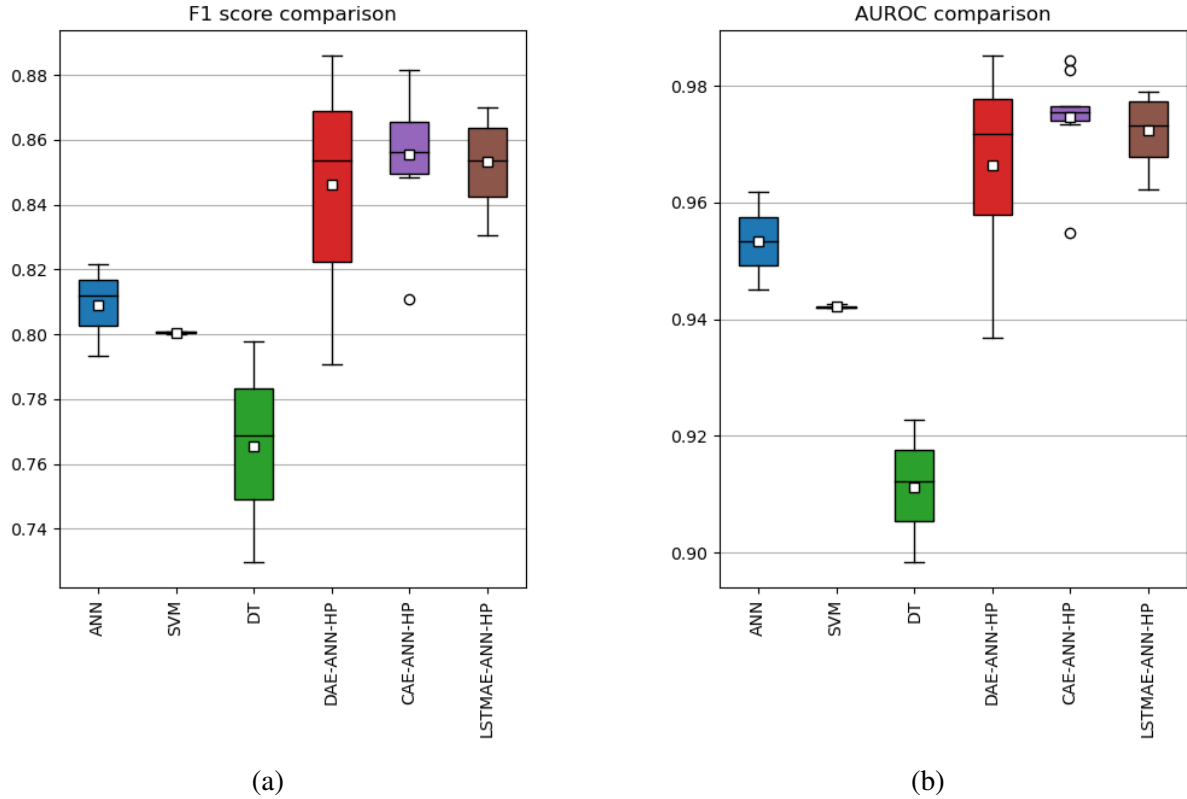


Figure 6.8: Comparison of generalization performances between all 6 experimented methods: (a) F1 score, (b) AUROC score.

as demonstrated by the largest IQR for both F1 score and AUROC score for DAE-ANN-HP. On the other hand, the use of a convolutional autoencoder yielded the highest average F1 and AUROC scores across all experiments. More importantly, the results across nine trials show the smallest variance in all three AE-ANN-HP experiments. Hence, it can be concluded that a convolutional autoencoder is the most suitable option for the AE-ANN-HP framework out of the three tested alternatives.

The performance gain of AE-ANN-HP framework is mainly attributed to the addition of the autoencoder degradation analysis component. The multi-objective fitness score metric allows for finding the most fitted autoencoder models, i.e., one that not only minimizes reconstruction error for no degradation samples but also maximizes the difference between reconstruction errors for the three health conditions, for generating latent space features and reconstruction errors. Consequently, for example, in experiment II with DAE-ANN-HP, the new enriched

dataset for health condition classification only contains 42 features (27 for latent space features, 1 for reconstruction error, 8 for statistical features, and 6 for time-derived features) in total, as opposed to 158 for the baseline ANN model in Experiment I. Yet the DAE-ANN-HP demonstrates better performance with less features.

The overall performance of AE-ANN-HP framework demonstrates that it is a promising approach for HVAC health prognostics with HVAC power consumption and outside temperature data. Nonetheless, the AE-ANN-HP framework is not without limitations. Some potential limitations of the framework are as follows:

- The AE-ANN-HP framework frames HVAC health prognostics into a multi-class classification task and its performance is evaluated on a synthetic dataset with 3 predetermined levels of health conditions. However, in a real world scenario, the degradation of an HVAC system would likely be more gradual. Hence, the framework may struggle to correctly classify the HVAC health condition when the input features fall near a decision boundary.
- As mentioned, the degradation of an HVAC system would likely be a gradual process. However, EnergyPlus does not provide the functionality to vary the fault intensity parameter gradually over time to simulate the power consumption of an HVAC system that degrades gradually. Hence, we make the decision to fix the degradation at three levels to simulate the corresponding HVAC behaviors.
- This framework does not consider other factors that may affect HVAC power consumption in addition to outside temperature, e.g., the building occupancy information, because the collection of these data are not as easy. This means that the framework may not perform as well in situations where the building occupancy level can fluctuate by a large amount.

Chapter 7

Conclusion & Future Work

7.1 Conclusion

HVAC systems accounts for a significant portion of global energy consumption every year. Effective maintenance of HVAC systems can not only reduce the operational cost of the buildings but also improve human comfort of the indoor environments. Data-driven predictive maintenance has been applied by researchers to help with HVAC diagnostics and prognostics. Most existing researches focus on large HVAC systems with internal IoT sensors capable of reporting operational data. However, these methods are not applicable to HVAC systems without internal IoT sensors, despite their significant presence in buildings around the globe.

Consequently, this thesis proposes AutoEncoder and Artificial Neural Network based HVAC Health Prognostics (AE-ANN-HP) framework, a novel data-driven HVAC predictive maintenance framework that classifies the HVAC health condition into no degradation (ND), moderate degradation (MD), and severe degradation (SD), based on its daily power consumption and outside temperature trends. AE-ANN-HP incorporates the use of an autoencoder to learn a generalized representation from the time series data such that the reconstruction errors produced by the autoencoder represent the relative degradation of the data samples. To achieve this desired behavior for the autoencoder, an exhaustive grid search is conducted to train the

autoencoder models with different combinations of hyperparameters. Then, a multi-objective fitness score is used to select the optimal autoencoder model that is able to minimize the reconstruction errors for the no degradation condition and separate the reconstruction errors for all three health conditions. Once the optimal autoencoder is selected, a new enriched dataset is formed using the outputs from the autoencoder for the final daily HVAC health condition classification.

The AE-ANN-HP framework is evaluated with the use of three different types of autoencoders, namely dense autoencoder (DAE), convolutional autoencoder (CAE), and long short-term memory autoencoder (LSTMAE). DAE is the simplest form of autoencoder, whereas CAE and LSTMAE have more complex structures that allow both to extract temporal features from the input data. The performances of these models are also compared to existing HVAC health prognostics methods including ANN, SVM, and DT models, evaluated on the same dataset. The best performing existing method is ANN, which yields a testing macro-averaged F1 score of 0.8094 and a testing macro-averaged AUROC score of 0.9456. The AE-ANN-HP models outperform the ANN model by an increase in macro-averaged F1 score of as high as 6.89% and an increase in macro-averaged AUROC score of as high as 3.51%. Within three AE-ANN-HP models, CAE-ANN-HP and LSTMAE-ANN-HP outperform the simpler DAE-ANN-HP model thanks to their better generalization abilities.

Overall, the AE-ANN-HP framework is capable of correctly classifying daily HVAC health conditions in the majority of the cases when paired with a LSTMAE. It proves that AE-ANN-HP is a promising approach for HVAC health prognostics using only daily HVAC power consumption and outside temperature trends. Because of the relative ease for collecting these two time series data, AE-ANN-HP has the potential to aid predictive maintenance decision making for HVAC systems without IoT internal sensors.

7.2 Future Work

Future work will mainly revolve around keeping on improving the framework's performance and the further validation of the framework. The future work include:

- All three AE-ANN-HP experiments still show tendency to consistently misclassify some health conditions to be more deteriorated than they actually are. Further studies are needed to investigate this behavior and attempt to further improve model performance. One potential direction may be to consider other classification models.
- As discussed, one of the limitation of the framework is that the framework may struggle to correctly classify a data sample that falls near the decision boundary. This problem may be mitigated by increasing the number of health condition classes or changing to a regression task with continuous space. Hence, these two approaches need further exploration.
- Experiment with other types of autoencoders. In this thesis, we evaluate AE-ANN-HP with DAE, CAE, and LSTMAE. However, there are other types of autoencoder structures that can also be applied to the framework. An example is the variational autoencoder structure, which has also seen success in the field of fault detection.
- The autoencoder's ability to distinguish between different data samples in different degradation classes may be further enhanced with contrastive representation learning, which is an emerging technique that allows an autoencoder to learn a latent space in which similar samples are close together and dissimilar samples are distanced. Hence, further exploration is needed to apply contrastive learning technique to autoencoder training of the framework.
- Investigate if and how to vary the fault intensity parameter in EnergyPlus fault models over time to simulate a more realistic degradation behavior of the HVAC system. This will mitigate the problem of only having fixed degradation condition for each simulation.

- Use a real HVAC dataset for further validation of the framework. Recall that the reason EnergyPlus is used in this thesis is because the real HVAC dataset are not labeled nor contain HVAC behaviors under significant degradation conditions. As data are still being collected, the real dataset will eventually contain enough data to be used for framework validation.

The proposed AutoEncoder and Artificial Neural Network based HVAC Health Prognostics (AE-ANN-HP) framework performs well to classify HVAC health condition from only daily power consumption and outside temperature trends. However, there is still room for improvements as have been discussed in this section.

Bibliography

- [1] IEA. Buildings. Technical report, IEA, 2022.

- [2] M. González-Torres, L. Pérez-Lombard, Juan F. Coronel, Ismael R. Maestre, and Da Yan. A review on buildings energy information: Trends, end-uses, fuels and drivers. *Energy Reports*, 8:626–637, 11 2022.

- [3] Niima Es-sakali, Moha Cherkaoui, Mohamed Oualid Mghazli, and Zakaria Naimi. Review of predictive maintenance algorithms applied to hvac systems. *Energy Reports*, 8:1003–1012, 11 2022.

- [4] R. Keith Mobley. *An Introduction to Predictive Maintenance*. Elsevier, 2002.

- [5] Gian Antonio Susto, Andrea Schirru, Simone Pampuri, Seán McLoone, and Alessandro Beghi. Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, 11:812–820, 6 2015.

- [6] Jovani Dalzochio, Rafael Kunst, Edison Pignatton, Alecio Binotto, Srijnan Sanyal, Jose Favilla, and Jorge Barbosa. Machine learning and reasoning for predictive maintenance in industry 4.0: Current status and challenges. *Computers in Industry*, 123:103298, 12 2020.

- [7] Tiago Zonta, Cristiano André da Costa, Rodrigo da Rosa Righi, Miromar José de Lima, Eduardo Silveira da Trindade, and Guann Pyng Li. Predictive maintenance in the industry

- 4.0: A systematic literature review. *Computers & Industrial Engineering*, 150:106889, 12 2020.
- [8] Yunli Wang, Chunsheng Yang, and Weiming Shen. A deep learning approach for heating and cooling equipment monitoring. In *IEEE International Conference on Automation Science and Engineering*, volume 2019-August, pages 228–234. IEEE Computer Society, 8 2019.
- [9] Jack C.P. Cheng, Weiwei Chen, Keyu Chen, and Qian Wang. Data-driven predictive maintenance planning framework for mep components based on bim and iot using machine learning algorithms. *Automation in Construction*, 112:103087, 4 2020.
- [10] Yassine Bouabdallaoui, Zoubair Lafhaj, Pascal Yim, Laure Ducoulombier, and Belkacem Bennadji. Predictive maintenance in building facilities: A machine learning-based approach. *Sensors*, 21:1044, 2 2021.
- [11] Saman Taheri, Amirhossein Ahmadi, Behnam Mohammadi-Ivatloo, and Somayeh Asadi. Fault detection diagnostic for hvac systems via deep learning algorithms. *Energy and Buildings*, 250:111275, 11 2021.
- [12] Chunsheng Yang, Burak Gunay, Zixiao Shi, and Weiming Shen. Machine learning-based prognostics for central heating and cooling plant equipment health monitoring. *IEEE Transactions on Automation Science and Engineering*, 18:346–355, 1 2021.
- [13] Huajing Sha, Peng Xu, Chonghe Hu, Zhiling Li, Yongbao Chen, and Zhe Chen. A simplified hvac energy prediction method based on degree-day. *Sustainable Cities and Society*, 51:101698, 11 2019.
- [14] Nabil Nassif. Impacts of air filters on energy consumption in typical hvac systems (sa-12-c010). In *ASHRAE Transactions*, volume 118, 01 2012.
- [15] Charu C. Aggarwal. *Neural Networks and Deep Learning*. Springer Cham, 2018.

- [16] Josh Patterson and Adam Gibson. *Deep Learning A Practitioner's Approach*. O'Reilly, 2017.
- [17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [18] J. K. Chow, Z. Su, J. Wu, P. S. Tan, X. Mao, and Y. H. Wang. Anomaly detection of defects on concrete structures with the convolutional autoencoder. *Advanced Engineering Informatics*, 45:101105, 8 2020.
- [19] H. D. Nguyen, K. P. Tran, S. Thomassey, and M. Hamad. Forecasting and anomaly detection approaches using lstm and lstm autoencoder techniques with the applications in supply chain management. *International Journal of Information Management*, 57:102282, 4 2021.
- [20] Ruiqi Tian, Luisa Liboni, and Miriam Capretz. Anomaly detection with convolutional autoencoder for predictive maintenance. In *2022 9th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, pages 241–245. IEEE, 11 2022.
- [21] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6791 LNCS:52–59, 2011.
- [22] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852. PMLR, 2015.
- [23] Sajjad Abdoli, Patrick Cardinal, and Alessandro Lameiras Koerich. End-to-end environmental sound classification using a 1d convolutional neural network. *Expert Systems with Applications*, 136:252–263, 12 2019.

- [24] Yang Yu, Chaoyue Wang, Xiaoyu Gu, and Jianchun Li. A novel deep learning-based method for damage identification of smart building structures. *Structural Health Monitoring*, 18(1):143–163, 2019.
- [25] Samir S. Yadav and Shivajirao M. Jadhav. Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data*, 6:1–18, 12 2019.
- [26] Jose Dolz, Karthik Gopinath, Jing Yuan, Herve Lombaert, Christian Desrosiers, and Ismail Ben Ayed. Hyperdense-net: A hyper-densely connected cnn for multi-modal image segmentation. *IEEE Transactions on Medical Imaging*, 38:1116–1126, 5 2019.
- [27] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 5 2018.
- [28] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28, pages 1310–1318. PMLR, 6 2013.
- [29] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, 2016.
- [30] Mohammad Navid Fekri, Harsh Patel, Katarina Grolinger, and Vinay Sharma. Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network. *Applied Energy*, 282:116177, 1 2021.
- [31] Alaa Sagheer and Mostafa Kotb. Unsupervised pre-training of a deep lstm-based stacked autoencoder for multivariate time series forecasting problems. *Scientific reports*, 9(1):1–16, 2019.

- [32] Drury B. Crawley, Linda K. Lawrie, Frederick C. Winkelmann, W. F. Buhl, Y. Joe Huang, Curtis O. Pedersen, Richard K. Strand, Richard J. Liesen, Daniel E. Fisher, Michael J. Witte, and Jason Glazer. Energyplus: creating a new-generation building energy simulation program. *Energy and Buildings*, 33:319–331, 4 2001.
- [33] Brijesh Pandey, Rangan Banerjee, and Atul Sharma. Coupled energyplus and cfd analysis of pcm for thermal management of buildings. *Energy and Buildings*, 231:110598, 1 2021.
- [34] A. Boyano, P. Hernandez, and O. Wolf. Energy demands and potential savings in european office buildings: Case studies based on energyplus simulations. *Energy and Buildings*, 65:19–28, 10 2013.
- [35] Debaditya Chakraborty and Hazem Elzarka. Early detection of faults in hvac systems using an xgboost model with a dynamic threshold. *Energy and Buildings*, 185:326–344, 2 2019.
- [36] Rongpeng Zhang and Tianzhen Hong. Modeling and simulation of operational faults of hvac systems using energyplus. Lawrence Berkeley National Laboratory, 8 2016.
- [37] Amir Ebrahimifakhar, Adel Kabirikopaei, and David Yuill. Data-driven fault detection and diagnosis for packaged rooftop units using statistical machine learning classification methods. *Energy and Buildings*, 225:110318, 10 2020.
- [38] Riccardo Satta, Stefano Cavallari, Eraldo Pomponi, Daniele Grasselli, Davide Picheo, and Carlo Annis. A dissimilarity-based approach to predictive maintenance with application to hvac systems. *CGnal Research Papers Series*, 1 2017.
- [39] Antonio Gálvez, Alberto Diez-Olivan, Dammika Seneviratne, and Diego Galar. Fault detection and rul estimation for railway hvac systems using a hybrid model-based approach. *Sustainability 2021, Vol. 13, Page 6828*, 13:6828, 6 2021.

- [40] Haidar Hosamo Hosamo, Paul Ragnar Svennevig, Kjeld Svidt, Daguang Han, and Henrik Kofoed Nielsen. A digital twin predictive maintenance framework of air handling units based on automatic fault detection and diagnostics. *Energy and Buildings*, 261:111988, 4 2022.
- [41] Norman L Tasfi, Wilson A Higashino, Katarina Grolinger, and Miriam AM Capretz. Deep neural networks with confidence sampling for electrical anomaly detection. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 1038–1045. IEEE, 2017.
- [42] Howard Cheung and James E. Braun. Simulation of fault impacts for vapor compression systems by inverse modeling. part i: Component modeling and validation. *HVAC&R Research*, 19(7):892–906, 2013.
- [43] Howard Cheung and James E. Braun. Simulation of fault impacts for vapor compression systems by inverse modeling. part ii: System modeling and validation. *HVAC&R Research*, 19(7):907–921, 2013.
- [44] Alberto Fernández, Salvador García, Francisco Herrera, and Nitesh V. Chawla. Smote for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, 61:863–905, 4 2018.
- [45] Ying Yan, Jun Cai, Tao Li, Wan Zhang, and Liangliang Sun. Fault prognosis of hvac air handling unit and its components using hidden-semi markov model and statistical process control. *Energy and Buildings*, 240:110875, 6 2021.
- [46] Mangesh Basarkar, Xiufeng Pang, Liping Wang, Philip Haves, and Tianzhen Hong. Modeling and simulation of hvac faults in energyplus. Lawrence Berkeley National Laboratory, 11 2011.

- [47] PNNL. Building re-tuning training guide: Occupancy scheduling: Night and weekend temperature set back and supply fan cycling during unoccupied hours. Technical report, Pacific Northwest National Laboratory, 2012.
- [48] Salvador García, Julián Luengo, and Francisco Herrera. *Data Preprocessing in Data Mining*. Springer, 2015.
- [49] Dalwinder Singh and Birmohan Singh. Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, 97:105524, 2020.
- [50] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 11 2020.
- [51] Lutz Prechelt. Early stopping - but when? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7700 LECTURE NO:53–67, 2012.
- [52] Young Jun Yoo. Hyperparameter optimization of deep neural network using univariate dynamic encoding algorithm for searches. *Knowledge-Based Systems*, 178:74–83, 8 2019.
- [53] Yesi Novaria Kunang, Siti Nurmaini, Deris Stiawan, and Bhakti Yudho Suprpto. Attack classification of an intrusion detection system using deep learning and hyperparameter optimization. *Journal of Information Security and Applications*, 58:102804, 5 2021.
- [54] Jasbir Arora. *Introduction to optimum design*. Elsevier, 2004.
- [55] Alan J Thomas, Miltos Petridis, Simon D Walters, Saeed Malekshahi Gheytsi, and Robert E Morgan. Two hidden layers are usually better than one. In *Engineering Applications of Neural Networks: 18th International Conference, EANN 2017, Athens, Greece, August 25–27, 2017, Proceedings*, pages 279–290. Springer, 2017.

- [56] Donald Azuatalam, Wee Lih Lee, Frits de Nijs, and Ariel Liebman. Reinforcement learning for whole-building hvac control and demand response. *Energy and AI*, 2:100020, 11 2020.
- [57] Rohit Chandra, Krishnanand Kaippilly Radhakrishnan, and Sanjib Kumar Panda. Trans-active control of air-conditioning systems in buildings for participation in singapore’s demand response market through load curtailment. *Sustainable Energy, Grids and Networks*, 31:100742, 9 2022.
- [58] Drury B. Crawley and Linda K. Lawrie. Development of global typical meteorological years (tmyx). <https://climate.onebuilding.org/>, 2022.
- [59] Haibing Wu and Xiaodong Gu. Towards dropout training for convolutional neural networks. *Neural Networks*, 71:1–10, 11 2015.

Curriculum Vitae

Name: Ruiqi Tian

Post-Secondary Education and Degrees: University of Western Ontario
London, ON
2021 - 2023 M.E.Sc, Electrical and Computer Engineering
Collaborative Specialization in Artificial Intelligence

University of Western Ontario
London, ON
2017 - 2021 B.E.Sc, Software Engineering

Honours and Awards: Bill and Barbara Etherington Research Fellowship
2020

Related Work Experience: Graduate Teaching Assistant
University of Western Ontario
2021 - 2023

Publications:

Tian, R., Liboni, L., & Capretz, M. (2022, November). Anomaly Detection with Convolutional Autoencoder for Predictive Maintenance. In 2022 9th International Conference on Soft Computing & Machine Intelligence (ISCFMI) (pp. 241-245). IEEE.