Mississippi State University
Scholars Junction

Theses and Dissertations Theses and Dissertations

8-6-2021

Advanced Data Analytics Methodologies for Anomaly Detection in Multivariate Time Series Vehicle Operating Data

Morteza Alizadeh
mory.alizadeh7@gmail.com

Advanced data analytics methodologies for anomaly detection in multivariate time series vehicle operating data

By

Morteza Alizadeh

Approved by:

Junfeng Ma (Major Professor)
J. Edward Swan II (Minor Professor)
Linkan Bian
Mohammad Marufuzzaman
Wenmeng Tian
Bo Tang
Linkan Bian (Graduate Coordinator)
Jason M. Keith (Dean, Bagley College of Engineering)

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in Industrial and Systems Engineering
in the Department of Industrial and Systems Engineering

Mississippi State, Mississippi

August 2021

Name: Morteza Alizadeh

Date of Degree: August 6, 2021

Institution: Mississippi State University

Major Field: Industrial and Systems Engineering

Major Professor: Junfeng Ma

Title of Study: Advanced data analytics methodologies for anomaly detection in multivariate time series vehicle operating data

Pages of Study: 179

Candidate for Degree of Doctor of Philosophy

Early detection of faults in the vehicle operating systems is a research domain of high significance to sustain full control of the systems since anomalous behaviors usually result in performance loss for a long time before detecting them as critical failures. In other words, operating systems exhibit degradation when failure begins to occur. Indeed, multiple presences of the failures in the system performance are not only anomalous behavior signals but also show that taking maintenance actions to keep the system performance is vital. Maintaining the systems in the nominal performance for the lifetime with the lowest maintenance cost is extremely challenging and it is important to be aware of imminent failure before it arises and implement the best countermeasures to avoid extra losses. In this context, the timely anomaly detection of the performance of the operating system is worthy of investigation. Early detection of imminent anomalous behaviors of the operating system is difficult without appropriate modeling, prediction, and analysis of the time series records of the system. Data based technologies have prepared a great foundation to develop advanced methods for modeling and prediction of time series data streams.

In this research, we propose novel methodologies to predict the patterns of multivariate time series operational data of the vehicle and recognize the second-wise unhealthy states. These approaches help with the early detection of abnormalities in the behavior of the vehicle based on multiple data channels whose second-wise records for different functional working groups in the operating systems of the vehicle. Furthermore, a real case study data set is used to validate the accuracy of the proposed prediction and anomaly detection methodologies.

Key words: Time series data analytics, Operational behavior prediction, Unhealthy states detection

DEDICATION

This dissertation work is dedicated to my always encouraging wife, Shamim Fathi, and to my supportive parents, Gholamali Alizadeh and Soraya Barzan, who have been constant sources of love and encouragement in my entire life.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the following personnel for their valuable insights and generous data support: Syamala Srinivasan, Iain Dodds, Mark Villaire, Mark Pompetzki, and Douglas Bunker, from nCode Company; and Randy Jones, T.C. Falls, and Michael Hamilton from ISER, Mississippi State University.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Background

Anomaly detection is a group of techniques to identify the abnormalities or outliers in the given data set. These techniques are of great interest to diverse fields, including data mining and machine learning, and play an essential role in a wide range of real-world applications, such as systems health monitoring, medical care, credit card fraud, and intrusion detection. The abnormalities or the outliers are those points or regions whose patterns do not conform to the expected values. These are the significantly discordant objects compared with the rest and could be the exceptions, aberrations, or peculiarities depending on the application scenarios. Anomaly detection has been applied in a variety of data-rich domains such as high-dimensional data, uncertain data, streaming data, network data, and time series data. Among them, time series data anomaly detection is of specific interest due to its inherent nature of time. The temporal (time series data) anomaly detection analysis aims to identify the abnormal behaviors of the system over time. Anomaly detection is truly an application based technique. For example, an abrupt change or unusual pattern in the financial or stock market data needs to be captured to prevent occasional disruption of markets; anomalies in the DNA sequences or proteins provide information about genetic mutation or diseases.

Vehicles systems include diverse operational subsystems, called Functional Working Groups (FWGs). Each FWG is included in the vehicle system for a specific function, and the integration of all FWGs aims to achieve the designed performance of the vehicle. Therefore, failure of any FWG degrades the vehicle performance and results in cost and time losses. This study aims to analyze time-series operational data of the vehicles to reduce maintenance costs and extend the lifetime of the vehicles. Hence, we develop a pattern prediction based anomaly detection methods to model and predict the behaviors of the vehicle and recognize the anomalous states based on individual time series data channels or multiple time series data channels of different FWGs. These models predict and monitor the health status of the operating vehicle during time series and identify unhealthy time intervals. The identified anomalous behaviors of the vehicle represent the performance issues, deterioration, or usage anomalies that require further technical investigation.

1.2 Data Scope

The scope of data in this study includes 15 months of time series data recorded between 2013 and 2014. The vehicle data box collects data in the format of CDF files. These CDF files are converted to CSV files for further analysis. The parametric data used in this study were collected with 1HZ frequency and includes approximately 100 data channels. This data set includes multivariate second-wise records between January 01, 2013, and March 31, 2014. The data were collected with 1 HZ frequency when the vehicle was actively operating. The raw data set consists of 1,973,797 rows and 101 data channels. In other words, the vehicle got turned on for a duration of 1,973,797 seconds, which amounts to 548 hours. This indicates that the vehicle was not operating on a 24/7 basis. Dealing with this data set is challenging due to its high dimensionality, combined operation status, and missing data.

1.3 Research questions

This research aims to cover three main research challenges in analyzing the performance of the operating vehicle based on multiple time series data channels as follows:

1. How to recognize and predict the vehicle behavior pattern using time series operational channels data?

2. How to detect the failure-in-performance states in single channel time series data?

3. How to detect the failure-in-performance states in multiple channel time series data?

1.4 Dissertation organization overview

In Chapter 2, we present a hybrid prediction methodology for characterizing the complex and dynamical behavior of the vehicle during large scale time series vehicle operating data. This approach analyzes the performance of each FWG individually to predict their behavioral pattern and recognizes the anomalous states in the data channel. As complex time series data usually involves both linear and nonlinear patterns, neither conventional AutoRegressive Integrated Moving Average (ARIMA) model nor neural network model is adequate to individually recognize the patterns accurately. We propose a novel hybrid ARIMA-WANN approach by integrating the ARIMA model and Wavelet Autoencoder Neural Network (WANN) to accurately detect anomalies for large scale time series vehicle operating data. To recognize the dynamic patterns, this approach exploits the strength of the ARIMA model in recognizing linear autocorrelations and the flexibility of the feed-forward WANN model in capturing the nonlinear nature of complex time series. In the training of the proposed WANN model, the backpropagation algorithm, the most popular algorithm for training neural networks, is used. Furthermore, we apply the proposed hybrid methodology for predicting the patterns and identifying the abnormal states of each data channel of a real large scale multiple-channel time series data set including one-year second-wise records of 101 operational performance channels of a specific vehicle.

With the growing complexity of modern vehicle systems, the capability of modeling the behavior of different vehicle subsystems and predicting their forthcoming patterns become vital for decision makers to extend the vehicle's life cycle and control its maintenance costs. Using statistical and deep learning approaches, various hybrid models are evaluated in Chapter 3 to model the behavior of the vehicle subsystems, predict the future trend, and consequently assist to make appropriate maintenance decisions. In this study, Auto-Regressive Integrated Moving Average (ARIMA), Multilayer Perceptrons Neural Network (MLPNN), and Wavelet Neural Network (WNN) are used to develop several series hybrid models (i.e. ARIMA-MLPNN, ARIMA-WNN, MLPNN-ARIMA, and WNN-ARIMA) to model and predict the behavior of the operating subsystems. Moreover, a threshold-based anomaly detection method is developed for the early detection of abnormalities. The analytical results of the case study show that the WNN-ARIMA model outperforms other hybrid models. A threshold-based anomaly detection approach developed based on the residual errors of the WNN-ARIMA model can accurately capture the abnormal states of vehicle subsystems which could support the vehicle maintenance decision making.

In Chapter 4, the research objective is to develop a multivariate model and an anomaly detection method to detect abnormalities in the operational behavior of different subsystems and identify the unhealthy states of the vehicle. This model monitors the health status of different subsystems and quickly recognizes the timely abnormalities or unexpected patterns based on multiple time series data. Given the multivariate time series data sequences collected by a variety of sensors installed in the vehicle, we propose a Multi-Layer Long Short Term Memory (LSTM) network combined with an Autoencoder architecture (ML-LSTMAE) to monitor and predict the operations of different components of the vehicle. By learning the proper encoding-decoding scheme in the training data, One-Class Support Vector Machine (OCSVM) algorithm is used to analyze the reconstruction errors and build a support boundary. Utilizing the learned support boundary, the prediction errors in the test data are further analyzed to distinguish the healthy and unhealthy states. To validate the performance of the proposed behavior modeling and anomaly detection approach, the real vehicle system multivariate data set and a NASA bearing multivariate data set are applied. The results from both case studies confirm the high accuracy of the proposed ML-LSTMAE network for learning the latent normal behaviors of different subsystems. Subsequently, the OCSVM algorithm can correctly classify the healthy and unhealthy states for both case studies.

CHAPTER II

PREDICTION AND ABNORMAL STATE DETECTION WITH HYBRID ARIMA-WNN MODEL IN LARGE SCALE TIME SERIES VEHICLE OPERATING DATA

2.1 Introduction: Time series anomaly detection

Anomaly detection has been widely investigated across various domains such as manufacturing, lifeline systems, and telecommunication networks. Many prior pertinent parametric and nonparametric studies have been conducted to analyze anomalies using data driven approaches, including Neural Networks [217, 176, 61], Wavelet analysis [135, 119, 141], entropy-based methods [68, 156, 136, 22, 170, 169] and Bayesian networks [86, 14, 51, 144].

It would be fascinating and useful to apply the anomaly detection techniques in the vehicle operational data records, towards the goal of determining whether the vehicle is healthy or unhealthy during operation. This research aims to identify the abnormal states of the vehicle based on the time-series records of data collected by monitoring parametric channels of the vehicle system. Outliers fall outside the normal operational profile of the vehicle and could represent performance issues, deterioration, or usage anomalies that require further investigation. The presence of outliers can also identify issues with the source data quality or data collection procedures. Outliers will be recorded and presented for comparison with the standard vehicle operating conditions to identify potential issues for further review. The application of anomaly detection would be challenging in the vehicle data, not only because of the scale but also for the diversity. Because, the vehicle performs in different extremities, including different geographic locations. So, it is challenging to identify different outliers, corresponding to different environments.

2.2 Review on nonlinear and non-stationary models for time series prediction and anomaly detection

We survey various methods studied in the literature and categorize them based on how they have been applied for predicting real-world time series applications. Broadly speaking, literature methodologies can be categorized into two groups of parametric and non-parametric approaches depending on whether the predictors are given a pre-determined form or are constructed purely based on the data. Figure 2.1 on page 56 represents a broad classification of time series prediction models.

2.2.1 Parametric models

Parametric models account for explicit functions with a finite number of parameters, which describe the relationship between the input (e.g., the intrinsic variables and their autoregressive terms) and the output variables (e.g., the future values of the intrinsic variables). These model parameters are estimated after time series realizations.

2.2.2 Autoregressive methods

Many anomaly detection studies have been focused on time series forecasting problems. Autoregressive models are most widely studied due to their flexibility in analyzing time series data [43]. Among them, AutoRegressive Moving Average (ARMA) is one of the well-known used time series analysis methodologies developed by [26], which assumes a linear relationship between lagged variables. Although these linear models are limited to model stationary time series process and generally fail to model the nonlinear and non-stationary processes. Later, AutoRegressive Integrated Moving Average (ARIMA) method was proposed to address the limitation of ARMA in handling the non-stationary time trends [58]. ARIMA method has been extensively used for various anomaly detection applications [130, 206, 232]. [225] investigated the ARIMA models to detect platoon and mobility anomalies and design a two-step prediction model for diminishing the false alarms due to road curves. [147] studied the traffic characterization and abnormality detection in network management by applying ARIMA model and traditional Holt-Winters methods. [145] developed an ARIMA-based anomaly detection model to specify the traffic network behavior and recognize the traffic patterns. [101] studied the ARMA and ARIMA forecasting methods for detecting abnormalities in the electricity consumption of residential and non-residential consumers. They found the ARIMA models are more suitable due to their capability in handling non-stationary consumption behavior. [209] proposed an improved ARIMA algorithm to detect traffic abnormalities in wireless sensor networks. [186] developed an ARIMA-based anomaly detection method to monitor patients' activities in several closed ward hospitals. [90] applied ARIMA technique to detect anomalies in the information system data collected through regular vehicle sensors to efficiently score and rank drivers.

Furthermore, [184] extended the linear ARMA models to Threshold Autoregressive (TAR) models to incorporate nonlinearity. TAR models were successfully used for time series prediction in neuroscience and economics [207]. To address the abrupt switch in Self-Excited Threshold AR (SETAR) models [183], [181, 167] introduced Smooth Transition AR (STAR) models to capture smooth transitions between regimes. Later [124] applied multi-regime smooth transition in heterogeneous AR models for financial time series prediction. [19] studied multi-regime threshold models to specify regime transitions based on time and time series values. Toward incorporating nonlinearity as a part of the model, [113] decomposed nonlinear and non-stationary time series into orthogonal trend series and detail series, and then developed ARMA and TAR models for predicting each decomposed time series. For nonlinear prediction, [102] integrated AR models with a Hidden Markov Model (HMM) in which AR parameters switch in time based on a finite-state Markov chain realization. Although most of these models were limited to nonlinear and stationary time series.

2.2.3 Neural Network and neuro-fuzzy-type methods

Neural Networks (NNs) as nonlinear time series prediction models have been widely used in different applications such as manufacturing systems, finance, systems health monitoring, health care, energy grids, etc [118, 99, 73]. These models do not rely on prior linearity or nonlinearity assumptions. That is, they are capable of approximating any continuous function to any arbitrary precision [143].

Feed-forward Neural Network models (FNNs) parameterized with the back-propagation algorithm have been widely used for nonlinear time series predicting [105, 59]. However, these models outperform conventional statistical models such as regression and Box-Jenkins approaches, the dynamics of time series in these models are time-invariant. [18] developed a Self-Organizing Map Neural Network model (SOMNN) to improve the predicting accuracy of FNNs for nonlinear time series. FNNs have also been extended to Recurrent Neural Network models (RNNs) by incorporating recurrent feedback connections [46]. RNNs have been employed for nonlinear time series prediction in various fields [67, 103, 64, 56]. [159] integrated a RNN with particle filtering models for one-step prediction of the nonlinear signal patterns in ultra-precision manufacturing processes. [127] developed an RNN structure of nonlinear AR models for multi-step prediction of chaotic time series. Later, [74] studied a variant of RNNs, called Long Short-Term Memory (LSTM), to avoid the problem of vanishing or exploding gradient in traditional RNNs. Using the Piece-Wise linear degradation concept [72, 78] investigated the LSTM implementation for estimating the Remaining Useful Life (RUL). [188] developed an LSTM model by training on both censored instances and failed instances. In terms of mean squared error and score, this model outperforms LSTM models proposed by [228, 228, 69], which only trains on failed instances. Besides, [129] studied a dual-task LSTM model to simultaneously assess the degradation and predict the RUL. [197] applied LSTM for modeling long-sequence trends and gradient boosting regression. To predict RUL, [111] combined LSTM and feature augmentation technique, which augments the forward difference between the current and previous values of sensors. Additionally, conventional LSTM models were extended to bi-directional LSTM (BiLSTM) models, which connects two hidden layers with opposite directions to the same output. These models can learn the dependencies of sensor data in both forward and backward directions [168]. BiLSTM models have also been attempted to estimate the RUL at time series data [80, 194, 219].

Ensemble NNs [218, 104] such as Wavelet Neural Networks (WNNs) are another variant of NN that have been studied for nonlinear time series prediction. For instance, [104] investigated an ensemble nonlinear NN model for financial time series prediction. In this model, different weights were initially generated, and then Principal Component Analysis (PCA) was employed to choose appropriate NN ensembles. A wavelet NN model facilitates the learning process of new time series as well as the separating process of noises from relevant information, especially for those time series with low frequency [177]. This superiority of WNN models is achieved by combining the functional approximation advantage of FNNs and the strength of wavelet analysis to demonstrate the non-stationary patterns in time series data [230]. There is a growing interest in neuro-fuzzy models for various time series prediction applications [98, 138, 163, 95]. These models can express large nonlinear time series whose underlying physical relationships are not known.

2.2.4 SVM methods

SVM methods founded based on generalized regression models, such as Support Vector Regression (SVR) and Least-Squares Support Vector Machines (LS-SVMs) [175, 16]. [191] studied a Bayesian method to parameterize LS-SVM models for predicting financial time series data. Using an exponentially increasing regularization and exponentially decreasing tube size, [29] proposed a dynamic SVM model to recognize the structural changes in the data. [106] investigated LS and Radial Basis Function (RBF) predictor to develop a local SVM for chaotic time series prediction. They showed that local SVM models with local LS- and RBF-based predictors result in higher accuracy for long-term prediction. [34] developed a reconstructed training-set SVM (RTS-SVM) to classify high-noise time series data in which the roulette cooperative coevolution algorithm (R-CC) is applied to optimize the parameters of RTS-SVM.

Hybrid SVM models have also been studied for predicting time series data. Applying a wavelet kernel to approximate arbitrary nonlinear functions, [221] proposed a wavelet SVM model. Assuming that most recent observations are most informative, [96] developed a fuzzy SVR model for predicting non-stationary time series data. This model diminishes over-fitting and computational costs compared to traditional SVR. [229] combined the Particle Swarm Optimization (PSO) and

SVM to accelerate the parameter selection of the model. [12] studied the hybrid ARIMA-SVM method to decompose time series data into a linear part for ARIMA prediction and nonlinear part for SVM prediction. The results showed efficient prediction compared with the results given by conventional methods.

2.2.5 Hidden Markov methods

Most of the methods reviewed above involve batch data processing. That is, the model is fitted and updated intermittently using batches of time series data. Although the curse of dimensionality due to large data sizes, memory requirements, and computational effort limits these methods application for solving real-world problems. Various sequential (also known as online or recursive) prediction models such as Hidden Markov Models (HMMs) [157] have been studied to address this significant limitation. An HMM is a special class of mixture models in which the observed time series data Y_t is treated as a function of unobserved/hidden state vector S_t. Figure 2.2 on page 56 depicts the graphical representation of an HMM.

The prediction performance of HMMs is particularly sensitive to the order of the Markov property that is employed to represent the states. That is, an n-th order Markov process in one where S_t, given $S_{t-1}, \ldots, S_{t-n}$, is independent of S_j for $j < t - n$. HMM is an attractive approach for modeling the time series data generated from a discrete state space. In this approach, switching between the finite states follows a definite pattern [63]. Generally, state space models such as Kalman Filter (KF) and Particle Filter (PF) are classified as HMMs. However, [200] developed the Extended Kalman Filter (EKF) model to address the restrictive Gaussian and linearity assumptions in KF models, it still assumes a Gaussian posterior. [192] introduced the Unscented KF (UKF) model to overcome the Gaussian limitation of EKFs. Dual KF methods [133] and Expected Maximization-based algorithms for nonlinear state space models [60] were also introduced for predicting the nonlinear and non-stationary time series data. [65, 100] investigated PF models for structured approximation with Bayesian estimation in nonlinear time series prediction. PF models estimate the posterior density by generating discrete samples from the continuous state variables. Using Fuzzy C-Means (FCM) clustering and fuzzy integral, [109] proposed an HMM model to detect anomalies in multivariate time series. [149] developed a new HMM model to treat temporal dependencies as latent variables over which inference is made. A variety of research has been focused on developing HMMs for mapping the degradation pattern and predicting RUL [197, 203, 49]. For instance, [197] introduced a cluster-based HMM to predict RUL and [203] developed three HMMs for predicting RUL value and its lower and upper bounds.

2.2.6 Literature on hybrid methods

Among many factors that are significant for selecting the best time series prediction model, accuracy is the most well-known criterion. Improving prediction accuracy has been always the focus of decision-makers in various fields of time series analysis. It is widely accepted that combining different models or leveraging hybrid approaches can considerably improve the prediction accuracy or succeed in dealing with the limitations of single models [15]. That is because the underlying process of real data generation cannot be easily determined or the single models may not be able to appropriately identify the true data generation process. Moreover, it is believed that combining heterogeneous models or hybridization will result in lower generalization variance or error [94]. In other words, hybrid models will reduce the risk of using an inappropriate model with low prediction accuracy by combining several models. Recently, studies about combination techniques of prediction models have attracted plenty of attentions [70]. The combination models can be mainly classified into two categories: series and parallel models. Parallel combination models generate the hybrid predictions by combing the predicting results of single models; while series combination models separate a time series analysis into two main components: the first model analyzes one of the components of the time series in the first stage and then another component is modeled in the second stage based on the results obtained in the first stage. Series combination models, particularly linear/nonlinear combinations, are one of the most commonly used hybrid approaches for time series predicting in various applications.

Incorporating hybrid linear/nonlinear approaches have attracted extensive attentions in time series analysis since the early study of [215]. [40] proposed a series hybrid model to predict seasonal time series by combining Seasonal Auto-Regressive Integrated Moving Average (SARIMA) and Support Vector Machine (SVM) methods. [150] combined ARIMA with Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) to estimate and predict machine health states based on vibration signal. [140] developed a hybrid ARIMA-SVM model for short-term load prediction. [38] proposed a hybrid ARIMA-GARCH model to predict short-time traffic flows. [107] introduced a hybrid model based on ARIMA and Genetic Programming (GP) to forecast the financial time series. [178] studied a combination of the Auto-Regressive Moving average (ARMA) model with different types of GARCH models to model and predict solar radiation. [11] constructed a hybrid ARIMA-GARCH model to detect attacks (anomalies) in network traffic.

In many applications, proper integration of linear and nonlinear models would provide more accurate results than individual linear or nonlinear approach in predicting time series data [15].

Several studies developed hybrid approaches by combining ARIMA with other methods for prediction [115, 47, 11]. By incorporating ARIMA and Artificial Neural Network (ANN), [215] proposed a hybrid ARIMA–ANN model, which shows better performance than individual ARIMA and ANN models in Wolf's sunspot, Candian lynx and exchange rate time series data sets. [83] developed hybrid approaches to combine conventional Autoregressive based models (e.g., AR, ARMA, and ARIMA) and ANN techniques to capture the non-linear nature of complex time series data. [211] integrated ARIMA and Multilayer ANN (MLANN) for nonlinear prediction of vehicle traffic flows.

Artificial Neural network (ANN) is among the most significant and widely used nonlinear models for time series prediction [125]. Many literature research constructed hybrid linear/nonlinear models using various types of neural network. [161] developed a hybrid neural and fuzzy network for system modeling and time series prediction. [132] developed a series hybrid approach for volatility forecasting in financial markets by integrating ANN and GARCH models. [1] combined ANN with an improved shark smell optimization algorithm to predict solar powers. Due to the capability of ANN in capturing nonlinear relationships, ANN is frequently used in the prior studies as a part of hybrid models. Moreover, combining ARIMA and ANN models is one of the common hybrid methods for time series prediction. Since the real-world systems barely reveal pure linear or nonlinear patterns, ARIMA and ANN are not individually appropriate to analyze both linear and nonlinear patterns together. Thus, combining these models can be an efficient approach to model real-world problems. Many recent studies constructed various ARIMA-ANN models by combining ARIMA with Multilayer Perceptrons Neural Network (MLPNN)[13, 196], Elman's recurrent neural networks [5], radial basis function neural network [171], and probabilistic neural networks [93] for time series prediction. Besides, [53] combined the ARIMA model with MLPNN and explanatory variables (ARIMAX) to predict air quality in urban areas. By combining ARIMA with MLPNN and Support Vector Regression (SVR), [54] presented two hybrid systems ARIMA-MLPNN and ARIMA-SVR for time series forecasting which also determines a suitable function for gathering the linear and nonlinear prediction components. In addition, [48] introduced a Particle Swarm Optimization (PSO) algorithm that searches for the best parameters of the linear and nonlinear components in a series of ARIMA-ANN prediction models. The ANN-ARIMA models have been studied for time series forecasting [162] and short-term traffic flow prediction [212]. [94] compared the performance of ARIMA-ANN and ANN-ARIMA for predicting stock prices. More recently, [71] evaluated four different hybrid models ARIMA–SVM, ARIMA–MLP, SVM–ARIMA, and MLP–ARIMA to distinguish better sequences in time series prediction.

On the other hand, many studies applied Wavelet Basis Functions (WBFs) as a transformation function in the hidden layer of standard ANN model [224] to take the self-organizing benefits of ANN and time-frequency properties of WBF [153, 180]. [199] pointed out that wavelet neural network (WNN) outperforms the conventional ANN for time series prediction due to its wavelet activation function in the hidden neurons. WNNs were employed in diverse time series prediction and anomaly detection domains such as renewable energy resources prediction [55, 164, 172, 198], stock price forecasting [108, 25], and network intrusion detection [116, 6, 82]. By leveraging wavelet activation function in the hidden neurons of WNN, several prior studies combined the ARIMA and WNN to improve the prediction accuracy. For instance, [81] constructed a multi-scale decomposition and reconstruction approach by combining Multi-Resolution Analysis (MRA), WNN, and ARIMA to predict real-time traffic behaviors. [160] proposed a hybrid ARIMA-WNN model to predict wheat yield time series data. [154] presented a hybrid ARIMA-WNN model to determine the performance status of the cloud environment and detect short-term performance anomalies. [220] combined ARIMA, WNN, and Improved Empirical Mode Decomposition (IEMD) to predict short-term electricity loads in power systems.

WNNs have been used in various applications including time series prediction [30, 45, 41], nonlinear modeling and approximation [201, 84, 24], and classification [151, 87]. [165] presented a hybrid method combines wavelet map patterns and supervised multilayer ANN to detect faults in rotating machinery. [190] combined discrete wavelet transform (DWT) and ANN to detect high impedance faults in distribution network. [88] integrated ANN, WNN, and Hilbert transform to detect short and long-term abnormal patterns. More recently, [198] combined WNN with classification technique and two detection strategies to identify anomalies in the ocean fix-point observing time series data.

Inspired by recent studies in anomaly detection of time series data and due to non-stationary of the proposed large scale vehicle operating data set, we developed a novel hybrid approach that combines ARIMA and WNN to predict behaviors and detect unhealthy states of the operating vehicles over the planning horizon. This method incorporates an ARIMA model in the first stage to construct the linear component of prediction and WNN in the second stage to generate the nonlinear component of prediction based on the results obtained from the first stage. Then, the performance of the proposed hybrid model for predicting the behavior of an operating vehicle is compared with the single ARIMA model as well as the single WNN model to validate the usability. Finally, a time series anomaly detection strategy is developed to recognize the unhealthy states of the vehicle by thresholding the relative prediction errors.

2.3 Prediction modeling and anomaly detection methodology

Modern vehicle systems become increasingly complicated due to the many precise and complex subsystems installed to satisfy customized requirements. The complexity creates many severe reliability problems that need to be recognized as early as possible to reduce the operating cost and extend the life cycle. Early detection of faults in the vehicle operating system is a highly significant research domain to sustain full control of the system because anomalous behaviors usually pose performance loss for a long time before detecting them as critical failures. In other words, the vehicle operating system exhibits degradation when failure begins to occur. Indeed, multiple presences of the performance failures in the vehicle system are not only the anomalous behavior signals but also show that taking maintenance actions to keep the system performance is vital. Maintaining a vehicle system in the nominal performance for the lifetime with the lowest maintenance cost is extremely challenging. It is important to be aware of imminent failure before it arises and implements the best countermeasures to avoid extra losses. In this context, the timely detection of the abnormal performance of the vehicle's operating system is worthy of investigation. Early detection of imminent anomalous behaviors of the vehicle operating system is difficult without appropriate modeling, predicting, and analyzing the time series records. This study focuses on developing a pattern prediction method to model and predict the behavior of the vehicle and recognize the unhealthy states based on multiple time series data recorded by different sensors of the vehicle. This model predicts and monitors the health status of the operating vehicle and identifies unhealthy time intervals. The identified anomalous behaviors of the vehicle represent the performance issues, deterioration, or usage anomalies that are worthy of further investigation.

In this research, we aim to design a prediction model to differentiate the healthy and unhealthy states of the vehicle using a large scale of time series data set. The vehicle system can be broken down into several FWGs. A vehicle is defined to be unhealthy if one or more FWGs are unhealthy. The healthy/unhealthy condition of each FWG will be determined based upon the parametric data channels. The parametric data channels will be analyzed and results will be mapped into FWGs to determine whether the vehicle is healthy or not.

We design a temporal anomaly detection framework by developing a hybrid prediction model over an offline multiple-channel time series data set. In this framework, the assumption of *temporal continuity* plays a significant role in modeling the behavior of the data channels [2]. This assumption states that the patterns in the data are not expected to change abruptly unless there are abnormalities in the data. Thus, the goal is to detect sudden changes in the trends of the underlying operational data channels. These sudden changes in the time series values are identified as anomalies and exhibit a lack of continuity with respect to their immediate data history.

2.3.1 Hybrid time series prediction model

This research aims to develop a hybrid prediction model for monitoring the behavior of an operating vehicle and alerting unhealthy states by analyzing the time series data of the operating vehicle. The vehicle system can be decomposed into several functions such as engine, transmission, and fuel systems. A vehicle is defined to be unhealthy if at least one of its functions is at unhealthy operating states. The healthy/unhealthy condition of each functions can be determined based on different time series data channels. In other words, the performance of various functions of the vehicle is analyzed to determine whether the vehicle is in a healthy state or not. Considering the *temporal continuity* role in modeling the behavior of the data channels, the goal of this research is to detect sudden changes in the trends of the underlying operational data channels.

In this section, we presented a hybrid anomaly detection model to characterize the stochastic process of each time series data channel. We start by modeling the linear relationship within the time series data channels using the ARIMA model. However, this model is only able to capture the linear correlations in stationary time series data, there are a lot of cases where the time series is not stationary. Hence, it is necessary to develop a robust prediction model to deal with non-stationary real data sets with nonlinear correlations. Due to the flexibility of ANNs in modeling nonlinear relationship, we incorporated a new class of neural networks, WNN, to tackle the nonlinear limitation of the ARIMA model. Thus, the proposed hybrid model utilizes the capability of the ARIMA model for predicting the unseen patterns without labeled historical data and flexibility of WNN in analyzing different variations of time series data. This research develops a novel hybrid prediction model by integrating the ARIMA model with WNN. Figure 2.3 on page 57 represents the broad representation of the proposed hybrid approach.

2.3.2 ARIMA model

In this section, we developed an ARIMA model to fit a linear model to each time series channel data based on several past observations and random errors. ARIMA models are the most general class of models for forecasting future values of time series by using historical data and random errors. These models consists of two main components: Auto Regressive (AR) and Moving

Average (MA). To fit a linear model to the time series data channel $c \in \mathcal{C}$, an ARIMA model can be formulated as follows. The general ARIMA process can be illustrated as below:

$$\phi(B)(1 - B)^d(x_t^c - \mu) = \theta(B)\varepsilon_t \quad \forall\, t \in \mathcal{T},\ c \in \mathcal{C} \tag{2.1}$$

where, x_t^c is the actual value of data channel $c \in \mathcal{C}$ in time $t \in \mathcal{T}$ and the white noise $\varepsilon_t \approx iid(0,\sigma^2)$. $\phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i$ and $\theta(B) = 1 - \sum_{j=1}^{q} \theta_j B^j$ stand for the polynomial functions of the backshift operator B with degree p and q, respectively. Moreover, ϕ_i, $i = 1, 2, \ldots, p$, and θ_j, $j = 1, 2, \ldots, q$ are the model parameters in which integers p and q are often referred to the orders of the model and d refers to as the order of differencing. In other words, order p refers to the number of lags of x_t^c to be used as regressors and order q indicates the number of lagged prediction errors for making prediction in ARIMA model.

According to Box and Jenkins [214] methodology, ARIMA modeling approach consists of three main steps: model identification, parameter estimation, and diagnostic checking. These steps are briefly described as follows:

1. Model identification: This step looks for the actual values of the number of auto-regressive terms (p), the number of moving average terms (q), and the number of differencing operations (d). To identify the order of the ARIMA models, Box and Jenkins [214] introduced the AutoCorrelation function (ACF) and the Partial AutoCorrelation function (PACF) of the sample time series data.These functions examine the time series data for determining the temporal correlation structure of the series; and in diagnostic checking step, the forecasting will be performed.

2. Parameter estimation: After defining the structure of the model ARIMA(p,d,q), this step estimates the model parameters which were identified in the previous step using the Ordinary Least Squared (OLS) method.

3. Diagnostic checking: The last step checks the adequacy of the constructed model. Indeed, this step checks whether or not the selected model appropriately model and predict the historical data. The best model structure is selected by using different diagnostic statistics such as the Sum of Squared estimate of Errors (SSE), Akaike Information Criterion (AIC), and residuals plots. The AIC is a measure of the quality of the model. This criterion rewards the goodness of fit of the model and includes penalty as the increasing function of the number of parameters in the model.

If the selected model does not adequately fit the sample time series data, a new ARIMA model will be constructed, and the three described steps will be repeated until the best model structure is found.

2.3.2.1 Autoregressive and Moving Average orders selection

After stationarizing time series data, we can testify the autocorrelation in the time series data using Ljung-Box Q-statistic [117]. This test is used to check if there is no autocorrelation existed in the time series data set. In this test, the rejection of null hypothesis exposes the dependency of sequential time series data and then the existed autocorrelation can be used to develop the prediction model.

The ACF and PACF are useful statistical tools for measuring the correlations between the current and earlier time series data and errors. These functions can be used to determine whether the autoregressive (AR) or moving average (MA) terms are necessary to realize the autocorrelation of the time series data. The ACF illustrates the coefficients of correlation between the values of the series and their lag values, while the PACF indicates the coefficients of correlation between the values of the series and their lags after removing the effect of any correlations due to the terms at shorter lags. As different channels in time series data set create different ACF and PACF graphs, it needs to select optimal ARIMA models.

2.3.2.2 ARIMA models comparisons

To determine the optimal seasonal and non-seasonal parameters for the ARIMA model to fit the time series data, we need to evaluate the performance of various models. Three significant criteria including (1) *p-value*, (2) error sum of squares (SSE), and (3) Akaike information criterion (AIC) are used to assess the performance of the models. The *p-value* criterion determines whether there is any autocorrelation in the residuals of the ARIMA model. The SSE and AIC criteria are formulated as below:

$$SSE = \sum_{t=1}^{|\mathcal{T}|}(x_t^c - \hat{x}_t^c)^2, \quad \forall c \in C \tag{2.2}$$

$$AIC = |\mathcal{T}|\big(ln(\frac{2\pi SSE}{|\mathcal{T}|}) + 1\big) + 2k \tag{2.3}$$

where $|\mathcal{T}|$ and $k = p+q$ show the length of time series and the number ofterms estimated by the ARIMA model, respectively. The SSE criterion shows the squared value of the prediction residuals for the model. The AIC criterion assigns credits to the models which reduce aggregated error, while gives penalties to the models that increase error [3]. Thus, among the candidate models, a model with a large *p-value* and relevant small AIC and SSE values will be selected to fit the time series data. When an appropriate model is fitted and its parameters are determined, the residuals of the model's prediction will be taken to test the fitness to the time series data. Several approaches are applied to determine whether the residuals are independent and normally distributed. To evaluate the independence of the residuals, we plot the values of the residual autocorrelation function (RACF) against the lag numbers. If the ARIMA model is determined with proper parameters, the estimated autocorrelations of the residuals are uncorrelated and approximately normally distributed with the center of zero. The Ljung–Box–Pierce Q-statistic is used to test whether the autocorrelations of the residuals are white noise, which means the model is appropriate.

2.3.3 Wavelet Neural Network Model

WNN model is a new class of standard ANN models that involves wavelet analysis in the prediction model by incorporating WBF as a transformation function in the hidden layer of the neural network. Wavelet analysis is a powerful tool for analyzing various time series data [7]. [224] proposed WNN to address a series of drawbacks in standard feed-forward neural networks including random weight initialization, local minima, and model complexity. One of the significant strengths of WNN is its capability of estimating the nonlinear processes with limited or no information on processes. As a generalized radial basis function networks, the WNN model is composed of three layers: an input layer, a hidden layer, and an output layer. The input layer represents the time series of exploratory variables. The hidden layer contains the hidden units (i.e., wavelons) which transfer the input variables to translated and dilated versions of the mother wavelet. Finally, the output layer provides the estimations of the target values. The general structure of the proposed WNN model is presented in Figure 2.4 on page 57.

For each time series data channel $c \in C$, the input layer of WNN is fed with past lagged values of actual data $(x^c_{t-1}, x^c_{t-2}, \ldots, x^c_{t-\rho})$ $\forall t = \rho + 1, \ldots, |\mathcal{T}|$ as an input vector to predict the future value x^c_t. The hidden layer interacts between the input layer and the output layer with m nodes. The m nodes of this layer are connected to the single output node of the output layer. Thus, the input layer of the proposed WNN model consists of ρ nodes that are connected to m nodes of the hidden layer and to the single node of the output layer. Finally, the output layer predicts the future values of the time series. Since the one-step-ahead forecasting strategy is considered in this paper for predicting future values, the output layer of the proposed WNN model contains only one node. Therefore, $\hat{x}^c_t$ corresponds to t-th time series prediction of WNN for data channel $c \in C$ by involving ρ past time series data $(x^c_{t-1}, x^c_{t-2}, \ldots, x^c_{t-\rho})$ in the input layer and m wavelons in the hidden layer. Note that, however, deciding the number of lagged observations in the input layer, ρ, and the number of wavelons in the hidden layer, m, is vital for the proposed WNN architecture, no systematic rule exists to select these parameters, and the only way for determining the optimal ρ and m is trial and error [92].

Moreover, $\Omega^c = (\Omega^{c[0]}, \Omega^{c[1]}, \Omega^{c[2]})$ shows the connection weights of WNN for data channel $c \in C$ that are adjusted during the training phase. To perform well in presence of linearity, the weight set $\Omega^{c[0]} = (\Omega^{c[0]}_{t-1}, \ldots, \Omega^{c[0]}_{t-\rho})$ $\forall t = \rho + 1, \ldots, |\mathcal{T}|$ directly connects the lagged observations of input layer to the output unit. The weight set $\Omega^{c[1]} = (\Omega^{c[1]}_{t-i,j})$ $\forall i = 1, \ldots, \rho,\ j = 1, \ldots, m,\ t = \rho+1, \ldots, |\mathcal{T}|$ connects the input layer to the hidden layer in which $\Omega^{c[1]}_{t-i,j} = (\omega^{c[1]}_{(\tau)t-i,j}, \omega^{c[1]}_{(\vartheta)t-i,j})$ $\forall i = 1, \ldots, \rho,\ j = 1, ..., m,\ t = \rho + 1, \ldots, |\mathcal{T}|$consists of translation factors $\omega^{c[1]}_{(\tau)t-i,j}$ and dilation factors $\omega^{c[1]}_{(\vartheta)t-i,j}$. The weight set $\Omega^{c[2]} = (\Omega^{c[2]}_1, \Omega^{c[2]}_2, ..., \Omega^{c[2]}_m, \Omega^{c[2]}_{m+1})$ establishes the linear connections between the wavelons of hidden layer and the output unit. Each wavelon j of hidden layer is operated by a wavelet function as follows:

$$\Gamma_j(\chi^c) = \prod_{i=1}^{\rho} \psi(\frac{x^c_{t-i} - \omega^{c[1]}_{(\tau)t-i,j}}{\omega^{c[1]}_{(\vartheta)t-i,j}}) \quad \forall\, j = 1,..,m,\ t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C \qquad (2.4)$$

where ψ stands for mother wavelet. According to literature, three common mother wavelets are the Gaussian derivative, the second derivative of the Gaussian (i.e., so-called Mexican Hat), and the Morlet wavelet. Although choosing the mother wavelet depends on the case application and is not limited to the aforementioned functions. In this paper, the Mexican Hat function is used as a mother wavelet since it is proved to be useful and efficient in various applications [20, 24, 223]. Then, the proposed mother wavelet is defined by:

$$\psi(\theta^c_{t-i,j}) = (1 - (\theta^c_{t-i,j})^2)e^{-\frac{1}{2}(\theta^c_{t-i,j})^2} \forall\, i = 1, \ldots, \rho,\ j = 1,..,m,\ t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C \qquad (2.5)$$

where $\theta^c_{t-i,j} = \frac{x^c_{t-i} - \omega^{c[1]}_{(\tau)t-i,j}}{\omega^{c[1]}_{(\vartheta)t-i,j}}$. Thus, the proposed WNN model predicts t-th time series value of data channel $c \in C$ as follows:

$$\hat{x}^c_t = \Omega^{c[2]}_{m+1} + \sum_{j=1}^{m} \Omega^{c[2]}_j \Gamma_j(\chi^c) + \sum_{i=1}^{\rho} \Omega^{c[0]}_{t-i} x^c_{t-i} \quad \forall\, t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C \qquad (2.6)$$

Similar to standard neural networks, the connection parameters of the WNN model need to be initialized. The random initialization of the translation and dilation parameters may not be a suitable approach [142]. A wavelet is a rapidly decaying waveform with a finite duration, zero mean, and localized properties. Hence, a random initialization of parameters may lead to wavelons with zero values in hidden layer. Moreover, training methods such as gradient descent with random initialization are inefficient [222] due to their low training speed and potential local minima of loss function [152].

Leveraging the information that the wavelet analysis extracts from the input time series data, the wavelons parameters $\Omega^{c[1]}$ of the proposed WNN can be initialized in an efficient way. It should be noted that efficient initialization results in less training iterations and avoids local minima trap during the training phase. This paper used the translation and dilation initialization proposed by [224] as follows:

$$\omega^{c[1]}_{(\tau)t-i,j} = 0.5\ (M^c_{t-i} + N^c_{t-i}) \forall\ i = 1,\ldots,\rho,\ j = 1,..,m,\ t = \rho + 1,\ldots,|\mathcal{T}|,\ c \in \mathcal{C} \tag{2.7}$$

$$\omega^{c[1]}_{(\vartheta)t-i,j} = 0.2\ (M^c_{t-i} - N^c_{t-i}) \forall\ i = 1,\ldots,\rho,\ j = 1,..,m,\ t = \rho + 1,\ldots,|\mathcal{T}|,\ c \in \mathcal{C} \tag{2.8}$$

where M^c_{t-i} and N^c_{t-i} show the maximum and minimum values of input data x^c_{t-i} over training samples set $\mathcal{N}$ as follows:

$$M^c_{t-i} = max_{n=1,\ldots,\mathcal{N}}(x^c_{t-i,n}) \quad \forall\ i = 1,\ldots,\rho,\ t = \rho + 1,\ldots,|\mathcal{T}|,\ c \in \mathcal{C} \tag{2.9}$$

$$N^c_{t-i} = min_{n=1,\ldots,\mathcal{N}}(x^c_{t-i,n}) \quad \forall\ i = 1,\ldots,\rho,\ t = \rho + 1,\ldots,|\mathcal{T}|,\ c \in \mathcal{C} \tag{2.10}$$

where $|\mathcal{N}| = |\mathcal{T}| - \rho$ and $x^c_{t-i,n}$ stands for $(t-i)$-th lagged actual data of n-th training sample in data channel $c \in \mathcal{C}$. Since the initialization of the weights $\Omega^{c[0]}$ and $\Omega^{c[2]}$ is less important, they are randomly initialized in small values between 0 and 1. Once connection parameters are initialized, the WNN model begins training to determine the optimal weights that minimize the lost function. In this research, the ordinary BackPropagation (BP) algorithm is used to train the proposed WNN model. BP determines the percentage of contribution of each connection weight of the network to the training error. The training error of t-th sample in data channel $c \in \mathcal{C}$, e^c_t, is defined by the difference between the target value x^c_t and the WNN output $\hat{x}^c_t$. Then, the pairwise error E^c_t is used for network training as follows:

$$E^c_t = \frac{1}{2}(x^c_t - \hat{x}^c_t)^2 = \frac{1}{2}(e^c_t)^2 \quad \forall\ t = \rho + 1,\ldots,|\mathcal{T}|,\ c \in \mathcal{C} \tag{2.11}$$

Thus, the connection weights of the WNN model are trained to minimize the mean quadratic lost function as follows:

$$L^c = \frac{1}{|\mathcal{N}|}\sum_{t=\rho+1}^{|\mathcal{T}|} E_t^c = \frac{1}{2|\mathcal{N}|}\sum_{t=\rho+1}^{|\mathcal{T}|} (e_t^c)^2 \quad \forall\, c \in C \tag{2.12}$$

For each data channel $c \in C$, the WNN model is iteratively trained until a vector of connection weights $\Omega^c = (\Omega^{c[0]}, \Omega^{c[1]}, \Omega^{c[2]})$ that minimizes the cost function of equation (2.12) is found. At each training iteration κ, the derivative of the loss function with respect to the connection weights are computed to update the network parameters based on the following learning rule:

$$\Omega_{\kappa+1}^c = \Omega_\kappa^c - \gamma \frac{\partial L^c}{\partial \Omega_\kappa^c} + \eta(\Omega_\kappa^c - \Omega_{\kappa-1}^c) \quad \forall\, c \in C \tag{2.13}$$

where γ and η represent learning rate and momentum term that usually take values between 0 and 1. Note that the momentum term expedites the training phase and helps WNN to avoid oscillations. Therefore, the partial derivative of the loss function with respect to the network weight Ω^c at data channel $c \in C$ can be computed as follows:

$$\begin{aligned} \frac{\partial L^c}{\partial \Omega^c} = \frac{1}{2|\mathcal{N}|}\sum_{t=\rho+1}^{|\mathcal{T}|} \frac{\partial E_t^c}{\partial \Omega^c} = \frac{1}{2|\mathcal{N}|}\sum_{t=\rho+1}^{|\mathcal{T}|} \frac{\partial E_t^c}{\partial \hat{x}_t^c}\frac{\partial \hat{x}_t^c}{\partial \Omega^c} = \\ \frac{-1}{|\mathcal{N}|}\sum_{t=\rho+1}^{|\mathcal{T}|} (x_t^c - \hat{x}_t^c)\frac{\partial \hat{x}_t^c}{\partial \Omega^c} = \frac{-1}{|\mathcal{N}|}\sum_{t=\rho+1}^{|\mathcal{T}|} e_t^c \frac{\partial \hat{x}_t^c}{\partial \Omega^c} \forall\, c \in C \end{aligned} \tag{2.14}$$

The partial derivatives of the prediction $\hat{x}_t^c$ with respect to network weights $\Omega^{c[0]}, \Omega^{c[1]}$, and $\Omega^{c[2]}$ and input variables $x_{t-1}^c, \ldots, x_{t-\rho}^c$ are evaluated as follows:

1. Partial derivatives with respect to direct connections $\Omega^{c[0]} = (\Omega_{t-1}^{c[0]}, \ldots, \Omega_{t-\rho}^{c[0]})$:

$$\frac{\partial \hat{x}_t^c}{\partial \Omega_{t-i}^{c[0]}} = x_{t-i}^c \qquad \forall\, i = 1, \ldots \rho, t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C$$

2. Partial derivatives with respect to linear connections between wavelons and the output node, $\Omega_j^{c[2]}$:

$$\frac{\partial \hat{x}_t^c}{\partial \Omega_j^{c[2]}} = \Gamma_j(\chi^c) \quad \forall\, j = 1, .., m,\ t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C$$

3. Partial derivatives with respect to bias term $\Omega_{m+1}^{c[2]}$:

$$\frac{\partial \hat{x}_t^c}{\partial \Omega_{m+1}^{c[2]}} = 1 \quad \forall\, t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C$$

4. Partial derivatives with respect to translation parameters $\omega_{(\tau)t-i,j}^{c[1]}$:

$$\frac{\partial \hat{x}_t^c}{\partial \omega_{(\tau)t-i,j}^{c[1]}} = \frac{\partial \hat{x}_t^c}{\partial \Gamma_j(\chi^c)} \cdot \frac{\partial \Gamma_j(\chi^c)}{\partial \psi(\theta_{t-i,j}^c)} \cdot \frac{\partial \psi(\theta_{t-i,j}^c)}{\partial \theta_{t-i,j}^c} \cdot \frac{\partial \theta_{t-i,j}^c}{\partial \omega_{(\tau)t-i,j}^{c[1]}} = \Omega_j^{c[2]} \cdot \psi(\theta_{t-1,j}^c) \ldots \psi'(\theta_{t-i,j}^c)$$
$$\ldots \psi(\theta_{t-\rho,j}^c) \cdot \frac{-1}{\omega_{(\vartheta)t-i,j}^{c[1]}} = -\frac{\Omega_j^{c[2]}}{\omega_{(\vartheta)t-i,j}^{c[1]}} \psi(\theta_{t-1,j}^c) \ldots \psi'(\theta_{t-i,j}^c) \ldots \psi(\theta_{t-\rho,j}^c)$$
$$\forall\, i = 1, \ldots, \rho,\ j = 1, .., m,\ t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C$$

5. Partial derivatives with respect to dilation parameters $\omega_{(\vartheta)t-i,j}^{c[1]}$:

$$\frac{\partial \hat{x}_t^c}{\partial \omega_{(\vartheta)t-i,j}^{c[1]}} = \frac{\partial \hat{x}_t^c}{\partial \Gamma_j(\chi^c)} \cdot \frac{\partial \Gamma_j(\chi^c)}{\partial \psi(\theta_{t-i,j}^c)} \cdot \frac{\partial \psi(\theta_{t-i,j}^c)}{\partial \theta_{t-i,j}^c} \cdot \frac{\partial \theta_{t-i,j}^c}{\partial \omega_{(\vartheta)t-i,j}^{c[1]}} = \Omega_j^{c[2]} \cdot \psi(\theta_{t-1,j}^c) \ldots \psi'(\theta_{t-i,j}^c)$$
$$\ldots \psi(\theta_{t-\rho,j}^c) \cdot \frac{x_t^c - \omega_{(\tau)t-i,j}^{c[1]}}{(\omega_{(\vartheta)t-i,j}^{c[1]})^2} = \frac{\Omega_j^{c[2]}}{\omega_{(\vartheta)t-i,j}^{c[1]}} \cdot \frac{x_t^c - \omega_{(\tau)t-i,j}^{c[1]}}{\omega_{(\vartheta)t-i,j}^{c[1]}} \cdot \psi(\theta_{t-1,j}^c) \ldots \psi'(\theta_{t-i,j}^c) \ldots \psi(\theta_{t-\rho,j}^c)$$
$$= \frac{\Omega_j^{c[2]}}{\omega_{(\vartheta)t-i,j}^{c[1]}} \theta_{t-i,j}^c \psi(\theta_{t-1,j}^c) \ldots \psi'(\theta_{t-i,j}^c) \ldots \psi(\theta_{t-\rho,j}^c) = -\theta_{t-i,j}^c \frac{\partial \hat{x}_t^c}{\partial \omega_{(\tau)t-i,j}^{c[1]}}$$
$$\forall\, i = 1, \ldots, \rho,\ j = 1, .., m,\ t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C$$

6. Partial derivatives with respect to input variables x_{t-i}^c:

$$\frac{\partial \hat{x}_t^c}{\partial x_{t-i}^c} = \Omega_{t-i}^{c[0]} + \sum_{j=1}^m \Omega_j^{c[2]} \frac{\partial \Gamma_j(\chi^c)}{\partial \psi(\theta_{t-i,j}^c)} \cdot \frac{\partial \psi(\theta_{t-i,j}^c)}{\partial \theta_{t-i,j}^c} \cdot \frac{\partial \theta_{t-i,j}^c}{\partial x_{t-i}^c} = \Omega_{t-i}^{c[0]} + \sum_{j=1}^m \Omega_j^{c[2]} \cdot \psi(\theta_{t-1,j}^c)$$
$$\ldots \psi'(\theta_{t-i,j}^c) \ldots \psi(\theta_{t-\rho,j}^c) \cdot \frac{1}{\omega_{(\vartheta)t-i,j}^{c[1]}} = \Omega_{t-i}^{c[0]} + \sum_{j=1}^m \frac{\Omega_j^{c[2]}}{\omega_{(\vartheta)t-i,j}^{c[1]}} \psi(\theta_{t-1,j}^c) \ldots \psi'(\theta_{t-i,j}^c) \ldots$$
$$\psi(\theta_{t-\rho,j}^c) = \Omega_{t-i}^{c[0]} - \sum_{j=1}^m \frac{\partial \hat{x}_t^c}{\partial \omega_{(\tau)t-i,j}^{c[1]}}$$
$$\forall\, i = 1, \ldots, \rho,\ j = 1, .., m,\ t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C$$

After the initialization phase, the connection parameters of the WNN model are trained during the learning phase for approximating the target values. Finally, the training phase stops when one of the following criteria is met: the difference between cost functions of consecutive iterations be less than 10^{-5} or the maximum iteration of 1000 epochs.

2.3.4 Hybrid ARIMA-WNN model

This section presents the proposed hybrid approach that combines ARIMA and WNN models. It is considered that the time series data of each channel $c \in \mathcal{C}$ consists of a linear autocorrelation structure and a nonlinear component as follows:

$$x_t^c = \Upsilon_t^c + \Phi_t^c \quad \forall\, t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in \mathcal{C} \tag{2.15}$$

where Υ_t^c and Φ_t^c respectively show the linear and nonlinear components that are estimated from the data using the proposed hybrid ARIMA-WNN model. The idea of hybrid modeling arises from the fact that if a time series data is modeled by a linear model such as ARIMA, its prediction residuals will only contain nonlinear structure. Then, the nonlinear component of the time series can be modeled based on the linear model's residual errors. Hence, the proposed hybrid ARIMA-WNN model utilizes unique attributes and strengths of ARIMA and WNN models in exploring different patterns. Additionally, since realizing the characteristics of the data in real problems is almost difficult, modeling the linear and nonlinear components of time series sequentially by different models improves the overall prediction accuracy in practical uses. Thus, the linear part of the time series is computed by ARIMA model in the first stage as follows:

$$r_t^c = x_t^c - \hat{\Upsilon}_t^c \quad \forall\, t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in \mathcal{C} \tag{2.16}$$

where $\hat{\Upsilon}_t^c$ and r_t^c stand for the output and residual errors of the ARIMA model at time t of data channel $c \in C$, respectively. Then, the nonlinear part is modeled by the WNN in the second stage using the residuals of the first stage model. With ρ input nodes, the WNN model for the residuals is presented as follows:

$$\hat{\Phi}_t^c = f(r_{t-1}^c, r_{t-2}^c, ..., r_{t-\rho}^c) \quad \forall\, t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C \tag{2.17}$$

$$r_t^c = \hat{\Phi}_t^c + \varepsilon_t^c \quad \forall\, t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C \tag{2.18}$$

where f, $\hat{\Phi}_t^c$, and ε_t^c indicate the nonlinear function determined by WNN, the predicting value at time t, and the corresponding random errors. Note that if the model f does not fit well for the first stage residuals, the error term will not be necessarily random. Hence, configuring an appropriate WNN model is essential. Consequently, the combined prediction for time t in data channel $c \in C$ is computed as follows:

$$\hat{x}_t^c = \hat{\Upsilon}_t^c + \hat{\Phi}_t^c \quad \forall\, t = \rho + 1, \ldots, |\mathcal{T}|,\ c \in C \tag{2.19}$$

2.3.5 Unhealthy State Detection Strategy

In this research, the normal behavior of the time series in different data channels is modeled by ARIMA-WNN such that a significant deviation from this model is considered as the abnormal behavior. Given the actual observed data x_t^c and the predicted value $\hat{x}_t^c$, the residual error $\varepsilon_t^c = x_t^c - \hat{x}_t^c$ is used as a deviation metric to identify the unhealthy state of the vehicle at time t for data channel c. If the absolute value of error falls outside the pre-defined threshold, an abnormal behavior alert is issued. Using hypothesis testing, the maximum likelihood distribution of residual errors ε_t^c is determined. Since the residual error sequence follows a certain distribution (e.g., normal, lognormal, gamma, logistic), its probability distribution can be estimated by using the maximum likelihood estimation approach. Then, the classifying threshold T^c for recognizing the unhealthy states of the vehicle based on data channel c can be constructed [198] as follows:

$$T^c = \frac{1}{2}\left[\left|a + \ln(\frac{\alpha}{1-\alpha})b\right| + \left|a + \ln(\frac{1-\alpha}{\alpha})b\right|\right] \quad \forall\, c \in C \tag{2.20}$$

where a and b are the parameters of the probability distribution fitting for the residuals of different data channels. In case of normal distribution, these parameters are mean $a = \mu$ and standard deviation $b = \sigma$. For logistic distribution, $a = \mu$ and $b = s$ are the location and scale parameters. The classifying level α (e.g., 90%, 0.95%, 99%, etc.) indicates the probability to accurately predict the values of different data channels. Note that the proposed threshold is computed based on the classifying level instead of user-defined value. This method guides constructing the threshold in the form of predictive levels that requires no experiential knowledge or parameter measurements.

2.4 Operating Vehicle Multiple Channel Time Series Data Wrangling

2.4.1 Data Description and Challenges

The vehicle time series parametric data set used in this study includes approximately 100 data channels. This data set includes the time series records for the vehicle consisting of 381 days second-wise records with the time span of January 1, 2013 through March 31, 2014. The data were collected only when the vehicle got turned on with a frequency of 1HZ. The raw data consists of 1,973,797 rows and 101 columns. In other words, the vehicle got turned on for a duration of 1,973,797 seconds, which amounts to 548 hours. This indicates that the vehicle was not operating on a 24/7 basis. Dealing with this data set is challenging due to high dimensional data, combined operation status, and missing data. To test the proposed hybrid approach, we use a relatively small scale of the data set. This partial data set includes three-month records including 473,290 rows of 101 attributes as multiple time series data channels per second. These channels include vehicle speed, engine oil pressure, transmission gear, engine torque, and others. These parametric data channels are 'intermittent' and do not have continuous values since the vehicle is not operating with a 24/7 schedule. The data set must be broken into on/off cycles. The *on-cycle* accounts for a time period in which the vehicle was turned on and the data were collecting whereas the *off-cycle* shows that the vehicle was turned off during that time period and no data were collected. If there is a gap of 10 seconds or more, it is assumed that the vehicle was turned off and no data were recorded. An on-off cycle indicator is created and added as an extra channel to the parametric data. Thus, the time series data of this study is wrangled using below steps:

1. All data channels $c \in C$ with at least 20% missing values throughout all timestamps are excluded from the analysis. Thus, 21 data channels such as unsprung mass, roll angels, and relative speed of front axles are removed from the main data set.

2. All data channels $c \in C$ with a constant value or very low standard deviation are excluded from the analysis. Thus, 53 channels such as vehicle brake dynamic control, clutch switch, and brake switch are removed.

3. Let ϱ_t represents the value of t-th timestamp of data. Enumerating the time gaps between sequential on-cycle data sets, the main data is decomposed to multiple on-cycle subsets $\mathcal{K} = \{1, \ldots, |\mathcal{K}|\}$ as follows:

$$Y_k = \left\{x_t^c, \forall\, t \in \mathcal{T},\ c \in C \mid \varrho_t - \varrho_{t-1} < 10\right\} \quad \forall\, k \in \mathcal{K} \tag{2.21}$$

4. When the data is not correctly captured for data channel $c \in C$, a null value will be assigned to it. We referred to these null values as missing values. Generally, the null values are deleted from the data or can be imputed. Since the data is continuous within each on-cycle Y_k, these null values can be imputed. If null values are at the beginning or end of the cycle, they are replaced by the nearest non-null value. If they are in the middle of the cycle, they are imputed using a linear interpolation method. Imputation is used only for a maximum of 30 points in a row. If there are null values for more than 30 points in a row, other than 30 rows are not imputed and are defined as missing values or NaN's.

5. If there are two duplicate channels, the channel with a higher percentage of null values is removed.

6. Although the data is continuously collected during each on-cycle Y_k, it does not mean that the vehicle is continuously under driving status. In other words, evaluating the values of various data channels reveals the different operating status of the vehicle. To better analyze the behavior of the vehicle, it needs to recognize the possible operating status of the vehicle for each on-cycle data. By investigating many data cycles, four major status (i.e., Idle-normal, Idle-throttle, Idle-high, and driving) are identified for defining the different operating status of the vehicle. Therefore, each on-cycle data set Y_k is decomposed to Idle-normal Y_k^{IN}, Idle-throttle Y_k^{IT}, Idle-high Y_k^{IH}, driving Y_k^D and Other Y_k^O subsets. Using three significant data channels engine speed, vehicle speed, and accelerator pedal position, Table 2.1 on the following page represents the major operating status of the vehicle as follows:

Table 2.1

Operational status of the vehicle

Case	ES[a] (RPM)	VS[b] (MPH)	APP[c] (%)	State
1	(650 , 750)	< 0.1	< 3.0	Idle-Normal
2	-	< 0.1	≥ 3.0	Idle-Throttle
3	≥ 770	< 0.1	< 3.0	Idle-High
4	-	≥ 0.1	≥ 0.0	Driving

[a] Engine speed
[b] Vehicle speed
[c] Accelerator pedal position

where the other operational states are shown by the *Other* status. Case 1 represents a low-speed status for the engine of the vehicle which means the vehicle is turned on, but not moving. For instance, this case happens when the vehicle is turned on and stopping at a traffic light. In case 2, the vehicle is not moving, but the driver is pushing the accelerator pedal. For instance, when the vehicle gets started in the winter and the driver pushes the accelerator pedal to warm up the vehicle engine. In case 3, the vehicle is not moving but the engine speed is high. It happens when the driver is running PTO or driving a pump. In case 4, the vehicle is moving. Even though the pedal position can be 0.0%, the vehicle is still moving. For instance, the driver may be applying brakes or driving downhill.

The proposed pre-processing procedure incorporates appropriate techniques to decompose the on-cycle data sets to five different subsets to address the missing values, dimensional challenges, and diverse operating status of the vehicle. Since the attributes of multiple time series data in various subsets are different, independent predicting models are developed to fit each of the operational data subsets. These models predict the behavior and identify the unhealthy states of the vehicle for different data channels.

2.4.2 Channel data preparation

Applying the proposed pre-processing steps on the main data set reveals that not all of the data channels are helpful to model the behavior of the vehicle. Hence, the main data set is adjusted to include the most useful data channels. This procedure identifies the useless data channels which not only require more computational efforts but also decrease the accuracy of the predictions. Consequently, 23 time series data channels are selected as the most important channels to implement the proposed hybrid ARIMA-WNN model for predicting the behavior of the vehicle and detecting the unhealthy states.

Since the data set of each on-cycle Y_k consists of time series data of different operating status, it is necessary to recognize the time intervals of various status throughout the multiple data channels and individually analyze them with different prediction models. Hence, preparation steps are developed to decompose each on-cycle data k, $\forall k \in \mathcal{K}$ to various operating intervals Y_k^{IN}, Y_k^{IT}, Y_k^{IH}, Y_k^{D} and Y_k^{O}. Running the proposed time gaps identification steps resulted in $|\mathcal{G}| = 438$ gaps throughout the main data which indicates that the vehicle was turned on and subsequently turned off 438 times in three-month data records. Thus, $|\mathcal{K}| = 439$ on-cycle intervals were generated each of which with a duration of the time length between the tuning-on and the subsequent turning-off status of the vehicle.

Moreover, as the vehicle operates differently in various operating status, it is essential to divide each on-cycle interval into independent operating subsets based on different status of the vehicle.

Therefore, the whole on-cycle intervals are decomposed to $\left|Y_1^{IN}\right| + \ldots + \left|Y_{|\mathcal{K}|}^{IN}\right| = 113,882$ Idle-normal subsets, $\left|Y_1^{IT}\right| + \ldots + \left|Y_{|\mathcal{K}|}^{IT}\right| = 843$ Idle-throttle subsets, $\left|Y_1^{IH}\right| + \ldots + \left|Y_{|\mathcal{K}|}^{IH}\right| = 592$ Idle-high subsets, $\left|Y_1^{D}\right| + \ldots + \left|Y_{|\mathcal{K}|}^{D}\right| = 282,032$ Driving subsets, and $\left|Y_1^{O}\right| + \ldots + \left|Y_{|\mathcal{K}|}^{O}\right| = 5,942$ Other state subsets. Pseudocode 1 shows the proposed data preparation procedure.

Algorithm 1: Data preparation procedure

```
$Input : \chi \leftarrow \{x_t^c, \forall\, t \in \mathcal{T},\, c \in \mathcal{C}\}$
$\chi^c \leftarrow \{x_t^c, \forall\, t \in \mathcal{T}\}, \forall\, c \in \mathcal{C}$
$\chi^c \leftarrow \{\chi^c\} \neq \emptyset, \forall\, c \in \mathcal{C}$
$Y_k \leftarrow \{\}, \forall\, k \in \mathcal{K}$
Decompose $\chi$ to on-cycle subsets $Y_k, \forall\, k \in \mathcal{K}$
Remove $Y_k$ with standard deviation less than 0.1, $\forall\, k \in \mathcal{K}$
Decompose $Y_k$ to different operational sets $Y_k^{IN}, Y_k^{IH}, Y_k^{IT}, Y_k^{D}\ and\ Y_k^{O}, \forall\, k \in \mathcal{K}$
$Output : Y_k^{IN}, Y_k^{IH}, Y_k^{IT}, Y_k^{D}\ and\ Y_k^{O}, \forall\, k \in \mathcal{K}$
```

2.4.3 Hybrid ARIMA-WNN Model Implementation

After decomposing the on-cycle data series to appropriate operating subsets, the proposed ARIMA-WNN model are implemented on each operating subset to predict the behavior of the vehicle and identify the unhealthy states of each data channel $c \in \mathcal{C}$. The Pseudocodes 2 and 3 represent the steps of WNN and hybrid ARIMA-WNN models. After checking the stationarity and autocorrelation, ARIMA model is applied to realize the linear relationships of time series data for each channel. Then WNN algorithm is implemented to detect the remaining nonlinear relationships in the residual errors of the linear model. Finally, by fitting the maximum likelihood distribution, the residuals of the hybrid model are analyzed to recognize the unhealthy states of the vehicle.

To illustrate the linear behavior of data channels in different operating subsets, the proposed $ARIMA(p,d,q)$ model is implemented on the subsets that contain at least 25 number of observations [27]. This model leverages the dependency of each value of time series channel on its lagged values by $AR(p)$ and lagged residual errors by $MA(q)$. Using unit root tests (i.e., Augmented Dickey-Fuller, Dickey-Fuller GLS, Phillips-Perron, KPSS, and Variance Ratio), the stationarity of time series data at each channel is checked and d times of differencing are used to stationarize the non-stationary data channels. Using the astsa[1] package in R environment, the best fitted ARIMA model for different data channels of each operating subset is recognized. Then the proposed WNN model is programmed in python 3.7.3 environment to analyze the obtained residuals from the previous stage. Due to a large number of evaluating operating subsets and time series channels, results obtained for Fuel Rate (FR), Engine Torque (ET), and Injector Control Pressure (ICP) data channels of a driving subset with a time length of 22 minutes are presented in detail. The FR sensor collects the accurate measurements of the fuel consumption of the vehicle with the unit of U.S. Gallons per hour. The ET sensor records the percent torque of the engine. The ICP sensor monitors the fuel pressure going to the injectors with the unit of pounds per square inch. The data set is divided into an 80% train set and a 20% test set. The train set is applied as the background to train ARIMA-WNN model, and the test set is used to evaluate the performance of the models. Note that both training and test data are normalized by dividing with the standard deviation of the training data.

[1] Applied Statistical Time Series Analysis, https://cran.r-project.org/web/packages/astsa/index.html

Algorithm 2: WNN model

```
$Input : H_k^s, \forall\, k \in \mathcal{K},\ s \in \mathcal{S}, \mathcal{S} = \{IN, IH, IT, D, O\}$
$H_k^s \leftarrow \{E_k^{sc}, \forall c \in C\}, \forall\, k \in \mathcal{K},\ s \in \mathcal{S}$
$E_k^{sc} \leftarrow \{\varepsilon_{kt}^{sc}, \forall t \in \mathcal{T}\}, \forall\, c \in C,\ k \in \mathcal{K},\ s \in \mathcal{S}$
Initialize $\rho_k^{sc}$, $m_k^{sc}$, $\gamma_k^{sc}$, and $\eta_k^{sc}$ for $\chi_k^{sc}$, $\forall c \in C,\ k \in \mathcal{K},\ s \in \mathcal{S}$
Randomly initialize $\Omega_k^{sc} = (\Omega_k^{sc[0]}, \Omega_k^{sc[1]}, \Omega_k^{sc[2]})$ for $\chi_k^{sc}$, $\forall c \in C,\ k \in \mathcal{K},\ s \in \mathcal{S}$
for $k \leftarrow 1$ to $|\mathcal{K}|$ do
    for $s \in \mathcal{S}$ do
        if $nrows(Y_k^s) \geq 25$ then
            for $c \leftarrow C$ do
                while not Stop-Criterion do
                    for $t \leftarrow 1$ to $|\mathcal{T}| - \rho$ do
                        Compute $\Gamma_{kj}(\chi^{sc})$ based on equation (2.4) for $\chi_k^{sc}$
                        Generate prediction $\hat{x}_{kt}^{sc}$ based on equation (2.6)
                        Compute lost function $L_k^{sc}$ based on equation (2.12) for $\chi_k^{sc}$
                        Compute derivatives of weights $\frac{\partial L_k^{sc}}{\partial \Omega_k^{sc}}$based on equation (2.14)
                        Update weights $\Omega_k^{sc[0]}, \Omega_k^{sc[1]}, \Omega_k^{sc[2]}$ based on equation (2.13)
                    end
                end
            end
        end
    end
end
```

Algorithm 3: Hybrid ARIMA-WNN model

```
$Input : Y_k^s, \forall\, k \in \mathcal{K},\ s \in \mathcal{S}, \mathcal{S} = \{IN, IH, IT, D, O\}$
$Y_k^s \leftarrow \{\chi_k^{sc}, \forall\, c \in C\}, \forall\, k \in \mathcal{K}, s \in \mathcal{S}$
$R_k^{sc} \leftarrow \{r_{kt}^{sc}, \forall\, t \in \mathcal{T}\}, \forall\, c \in C,\ k \in \mathcal{K}, s \in \mathcal{S}$
$E_k^{sc} \leftarrow \{\varepsilon_{kt}^{sc}, \forall\, t \in \mathcal{T}\}, \forall\, c \in C,\ k \in \mathcal{K}, s \in \mathcal{S}$
$\alpha \leftarrow 0.05$
for $k \leftarrow 1$ to $|\mathcal{K}|$ do
    for $s \in \mathcal{S}$ do
        if $nrows(Y_k^s) \geq 25$ then
            for $c \leftarrow C$ do
                Perform $d$ times differencing to stationarize channel $\chi_k^{sc}$, if required
                Test autocorrelation in (stationarized) channel $\chi_k^{sc}$
                if autocorrelation exists in channel $\chi_k^{sc}$ then
                    Find candidate MA parameters $q$ by ACF
                    Find candidate AR parameters $p$ by PACF
                    Fit best ARIMA($p_k^{sc*}, d_k^{sc*}, q_k^{sc*}$) for $\chi_k^{sc}$ based on AIC and SSE
                    Fit WNN (Pseudocode 2) to $R_k^{sc}$ to find optimal $\rho_k^{sc}$, $m_k^{sc}$, $\gamma_k^{sc}$, and $\eta_k^{sc}$
                      for $\chi_k^{sc}$
                    Train WNN (Pseudocode 2) with optimal parameters for $\chi_k^{sc}$
                    Test WNN (Pseudocode 2) by one-step-ahead forecasting for $\chi_k^{sc}$
                    Fit ML distribution to $E_k^{sc}$ for $\chi_k^{sc}$
                    Construct classifying threshold $T_k^{sc}$ for $\chi_k^{sc}$
                    Issue an unhealthy alert once $\varepsilon_{kt}^{sc}$ exceeds $T_k^{sc}$ for $\chi_k^{sc}$
                end
            end
        end
    end
end
```

2.4.4 Results and discussion

The stationarized series, ACF, and PACF of the three FR, ET, and ICP data channels are shown in Figure 3.2 on page 96. Note that the blue lines in ACF and PACF plots represent the lower and upper bounds of the confidence interval. It means that falling of the correlation coefficient within the confidence bounds results in no significant lag whereas falling outside of the confidence interval indicates a significant lag. Figure 3.2 on page 96(a) shows that the ACF outside the confidence interval occurs at lags 2 to 4, and lags 11, 12, and 15 for the FR data channel. So the t-2, t-3, and t-4 as well as t-11, t-12, and t-15 are chosen as the relevant lags of prediction errors. Moreover, as PACF falls outside of the confidence bands at lags 2 to 6 and lags 11, 12, 16, 18, and 19, the t-2 to t-6 in addition to the t-11, t-12, t-16, t-18, and t-19 are selected as the relevant lags of previous observations. Analogously, for the ET data channel, the main relevant lags of x_t^c are t-2, t-3, and t-4 and the lags of prediction errors are t-2, t-3, t-4, and t-14. Also, the previous observations lags and the prediction errors lags respectively are t-3 to t-5, t-7, t-13, and t-21 as well as t-3 to t-5, t-6, t-9, t-13, and t-15 for the ICP data channel. Evaluating various combinations of the order parameters, Table 3.1 on page 80 represents the best-fitted models for different data channels. In this table, the best ARIMA models are selected based on low AIC and SSE values. The lower value of these metrics indicates a better fit for the ARIMA model. In other words, low AIC and SSE values show quality models with less complexity and lower training error.

Since ARIMA model leaves the nonlinear patterns in its prediction residuals, then WNN model can be used to discover these nonlinear relationships. Afterward, WNN model is trained for different dimensions of the input vector in the input layer $\rho \in [1, 25]$, the various number of neurons in the hidden layer $m \in [2, 10 \times \rho]$, and one output neuron. Among different structures

Table 2.2

Models evaluation for FR, ET, and ICP data channels

Channel	Best model	AIC	SSE
FR	ARIMA(4,1,2)	1848.81	357.97
ET	ARIMA(3,1,4)	1018.94	159.56
ICP	ARIMA(5,1,5)	901.24	141.32

of WNN model, the best model is obtained with the minimum training error. It should be noted that to evaluate the performance of the hybrid ARIMA-WNN model, WNN model is trained on the time series of different data channels as an individual prediction model as well as the second stage of the hybrid model. Table 2.3 on the following page shows the optimal values of ρ, m, γ, and η parameters to construct the best WNN models for different data channels FR, ET, and ICP.

Moreover, testing different probability distributions, the normal distribution and logistic distribution are considered to determine the probability distribution of the final prediction errors. Then, the confidence intervals 95%, 99%, and 99.5% are used for constructing different threshold levels.

2.4.4.1 Comparison of ARIMA, WNN, and ARIMA-WNN

In this section, the predictive capability of the proposed hybrid ARINA-WNN model is compared with ARIMA and WNN models. Three performance indicators Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Nash–Sutcliffe model Efficiency coefficient (NSE)

Table 2.3

Models evaluation for FR, ET, and ICP data channels

	WNN				Hybrid ARIMA-WNN			
Channel	ρ	m	γ	η	ρ	m	γ	η
FR	3	6	0.0001	0.1	8	14	0.01	0.1
ET	8	10	0.0001	0.1	3	6	0.01	0.0001
ICP	4	8	0.001	0.05	6	5	0.01	0.05

are employed to compare the prediction accuracy of ARIMA-WNN, ARIMA, and WNN models as follows:

$$MAE = \frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|} |\ x_t^c - \hat{x}_t^c\ | \times 100 \quad \forall\, c \in C \tag{2.22}$$

$$RMSE = \sqrt{\frac{1}{|\mathcal{T}|}\sum_{t=1}^{|\mathcal{T}|}\left(x_t^c - \hat{x}_t^c\right)^2} \quad \forall\, c \in C \tag{2.23}$$

$$NSE = 1 - \frac{\sum_{t=1}^{|\mathcal{T}|}\left(x_t^c - \hat{x}_t^c\right)^2}{\sum_{i=1}^{|\mathcal{T}|}\left(x_t^c - \overline{x}^c\right)^2} \quad \forall\, c \in C \tag{2.24}$$

where $\overline{x}^c$ is the average of time series data in channel $c \in C$.

Figure 2.6 on page 59, Figure 2.7 on page 60, Figure 2.8 on page 61 respectively show the expected values of FR, ET, and ICP channels estimated by ARIMA, WNN, and ARIMA-WNN models against the actual observation in test data. Moreover, table Table 2.4 on the following page presents the training and testing errors of the models for different channels FR, ET, and ICP. The WNN model improves the predictions of different data channels comparing to ARIMA model's performance. For example, the MAE of testing data in the FR channel is decreased from 36.368 for the ARIMA model to 20.794 for the WNN model. Also, the testing MAE of the ET data channel decreased from 24.682 in the ARIMA model to 9.045 in the WNN model. But,

Table 2.4

Performance evaluation of models for FR, ET, and ICP data channels

		ARIMA		WNN		ARIMA-WNN	
Channel		Train	Test	Train	Test	Train	Test
FR	MAE	39.153	36.368	19.873	20.794	18.864	15.204
	RMSE	0.59	0.534	0.269	0.269	0.227	0.179
	NSE	0.652	0.735	0.927	0.933	0.948	0.97
ET	MAE	19.911	24.682	7.751	9.045	9.814	6.313
	RMSE	0.394	0.399	0.101	0.119	0.137	0.078
	NSE	0.845	0.887	0.990	0.990	0.981	0.996
ICP	MAE	21.178	24.564	10.163	9.258	5.690	4.701
	RMSE	0.37	0.399	0.208	0.139	0.077	0.066
	NSE	0.863	0.843	0.956	0.981	0.994	0.996
Average	MAE	26.748	28.538	12.596	13.033	11.456	8.739
	RMSE	0.451	0.444	0.193	0.176	0.147	0.108
	NSE	0.787	0.822	0.958	0.968	0.974	0.987

both models underperform the proposed hybrid ARIMA-WNN model. Numerical results show that ARIMA-WNN model outperforms the component models ARIMA and WNN in terms of prediction accuracy. In other words, neither ARIMA model nor WNN model can individually capture the underlying trend of data on different channels. For instance, the RMSE of testing in the FR data channel is decreased from 0.534 and 0.269 for ARIMA and WNN models to 0.179 for ARIMA-WNN model. Likewise, the testing RMSE in the ICP data channel reduced from 0.399 in the ARIMA model and 0.139 in the WNN model to 0.066 in the ARIMA-WNN model. In addition, the NSE value of testing raises from 0.735 (FR), 0.887 (ET), and 0.843 (ICP) with ARIMA model and 0.933 (FR), 0.99 (ET), and 0.981 (ICP) with WNN model to 0.97 (FR), 0.996 (ET), and 0.996 (ICP) with ARIMA-WNN model.

Table 2.5 on the next page, Table 2.6 on page 47, and Table 2.7 on page 48 show the improvement percentages of different models for data channels FR, ET, and ICP, respectively. Besides, Table 2.8 on page 49 represents the average improvements of models over three data channels. Results indicate that ARIMA-WNN has overall yield better performance than WNN and ARIMA models. The RMSE indicator in Table 2.5 on the next page reveals that ARIMA-WNN model improves ARIMA predictions for 61.42% and WNN predictions for 33.35% in the test data of FR channel. Table 2.6 on page 47 shows that, in terms of RMSE, the ARIMA-WNN model can respectively improve 75.5% and 34.30% over than ARIMA and WNN models in the test data of ET channel. Table 2.7 on page 48 indicates the 75.5% and 29.92% RMSE improvements of the proposed hybrid model over ARIMA and WNN models in test data of ICP channel, respectively. Also, Table Table 2.8 on page 49 reports that ARIMA-WNN model can significantly improve the average RMSE of ARIMA and WNN models respectively for 73.33% and 32.66% in the test data. It can be seen that the proposed hybrid model enhances the average NSE of the ARIMA and WNN models for 19.92% and 1.82%, respectively, which suggests a model with more predictive skill. Furthermore, the average training errors over different channels is decreased by ARIMA-WNN model comparing with ARIMA and WNN models. Results verify that the proposed ARIMA-WNN model exploits the unique strength of ARIMA model in determining linear relationships and WNN model capability in detecting nonlinear patterns. Therefore, it would be more beneficial to capture the linear and nonlinear patterns separately and make the final prediction by combining the linear and nonlinear components which improve the overall predicting performance in different data channels.

Table 2.5

Improvement percentage of models for FR data channel

		ARIMA		WNN	
Indicator		Train	Test	Train	Test
MAE	WNN	49.24%	42.82%	0.0%	0.0%
	ARIMA-WNN	51.82%	58.19%	5.08%	26.89%
RMSE	WNN	54.35%	49.62%	0.0%	0.0%
	ARIMA-WNN	61.51%	66.42%	15.67%	33.35%
NSE	WNN	42.20%	26.91%	0.0%	0.0%
	ARIMA-WNN	45.42%	32.04%	2.27%	4.04%

2.4.5 Unhealthy state detection

The residual errors of the predictions generated by the proposed ARIMA-WNN model are then analyzed to detect the unhealthy states of the operating vehicle based on different data channels. Testing different probability distributions, the normal and logistic distributions are considered to fit the residual errors of the proposed hybrid model for different data channels. Using the maximum likelihood estimation method, the fitness of these distributions for different data channels is checked. Table 2.9 on page 50 represents the parameters and AIC indicator of these distributions and the corresponding anomaly detection thresholds with different classifying levels for data channels FR, ET, and ICP.

This table shows that comparing with the normal distribution, the logistic distribution has a better fit to the residuals of various data channels with lower AIC values. Correspondingly, the logistic threshold can be more effective in identifying the unhealthy states of the vehicle based on the residual distribution of different channels. Moreover, Table 2.10 on page 51 presents the

Table 2.6

Improvement percentage of models for ET data channel

		ARIMA		WNN	
Indicator		Train	Test	Train	Test
MAE	WNN	61.07%	63.35%	0.0%	0.0%
	ARIMA-WNN	50.71%	74.42%	-26.62%	30.21%
RMSE	WNN	74.39%	70.21%	0.0%	0.0%
	ARIMA-WNN	65.15%	80.43%	-26.52%	34.3%
NSE	WNN	17.14%	11.6%	0.0%	0.0%
	ARIMA-WNN	16.12%	12.21%	-0.88%	0.55%

false positive and false negative rates of these probability distributions when the classifying level increases from 90% to 99%.

Results for logistic threshold in table Table 2.10 on page 51 indicate that rising the classifying level from 90% through 99% increases the false negative rate of unhealthy states and decreases the false positive unhealthy rate in different data channels. Also, this table shows that rising the classifying level increases the false negative rate and decreases the false positive rate of the normal threshold for various data channels. In real applications like operating vehicle data, the actual unhealthy observations whose residual errors larger than the threshold are more significant to be classified correctly than those with residual errors smaller than the threshold. In other words, flagging the unhealthy states correctly is more important than alarming the healthy states incorrectly. Therefore, the 95% classifying level is chosen to construct the classifying threshold since it leads to a reasonable trade-off between the false negative and false positive unhealthy rates. Subsequently, it is expected that 5% of the healthy states will be classified as unhealthy states.

Table 2.7

Improvement percentage of models for ICP data channel

		ARIMA		WNN	
Indicator		Train	Test	Train	Test
MAE	WNN	52.01%	62.31%	0.0%	0.0%
	ARIMA-WNN	64.9%	73.16%	26.86%	28.8%
RMSE	WNN	43.84%	65.03%	0.0%	0.0%
	ARIMA-WNN	61.72%	75.5%	31.84%	29.92%
NSE	WNN	10.86%	16.30%	0.0%	0.0%
	ARIMA-WNN	13.57%	17.46%	2.44%	1.0%

The obtained results demonstrate that the threshold constructed by the model predictions can reasonably determine the classifying boundary between the healthy and unhealthy states of the operating vehicle. For instance, taking the 95% classifying level with the FR data channel, the normal distribution has a mean value μ=0.0004 and a standard deviation σ=0.269, and the logistic distribution has a location parameter of μ=0.006 and a scale parameter of s=0.143. Then, using equation 3.13, the respective thresholds are computed as T^{FR}_{normal}=0.792 and $T^{FR}_{logistic}$=0.42. For the ET data channel with the classifying level 95%, the normal threshold T^{ET}_{normal}=0.404 and the logistic threshold $T^{ET}_{logistic}$=0.203 are generated based on a a normal distribution with μ=0.006 and σ=0.137 and a logistic distribution with μ=0.02 and s=0.069, respectively. Similarly, for the ICP data channel with the same classifying level, the normal threshold T^{ICP}_{normal}=0.227 and the logistic threshold $T^{ICP}_{logistic}$=0.122 are computed according to a normal distribution with μ=0.003 and σ=0.077 and a logistic distribution with μ=0.001 and s=0.041, respectively. Figure 3.12 on page 111 depicts the normal and logistic probability distributions for the residual errors of the FR,

Table 2.8

Average improvement percentage of models

		ARIMA		WNN	
Indicator		Train	Test	Train	Test
MAE	WNN	52.91%	54.33%	0.0%	0.0%
	ARIMA-WNN	55.0%	67.17%	4.44%	28.11%
RMSE	WNN	57.30%	60.40%	0.0%	0.0%
	ARIMA-WNN	62.62%	73.33%	12.46%	32.66%
NSE	WNN	21.77%	17.77%	0.0%	0.0%
	ARIMA-WNN	23.28%	19.92%	1.24%	1.82%

ET, and ICP data channels. Obviously, the logistic distributions are narrower and more centered on the mean than the normal distributions. Thus, the classifying threshold of logistic distribution is suggested in the presence of a small persistent sequence of unhealthy states while the normal distribution threshold is more effective when the goal is to detect long-time anomalies.

Figure 2.10 on page 63 shows the unhealthy states flagged by the logistic and normal thresholds. It can be seen that the logistic threshold is more effective to identify the unhealthy states of the vehicle since these are sudden persistent changes in the operating status of different sensors of the vehicle. As shown in Figure 2.10 on page 63 (a), the logistic threshold can detect the sudden changes in the fuel rate of the operating vehicle. These changes are detected based on the residual errors of ARIMA-WNN model that are higher than the logistic threshold in Figure 2.10 on page 63 (b). Likewise, Figure 2.10 on page 63 (c) and Figure 2.10 on page 63 (d) represent the unhealthy states for the ET data channel. Figure 2.10 on page 63 (e) and Figure 2.10 on page 63 (f) indicate the unhealthy states in the ICP data channel using normal and logistic thresholds.

Table 2.9

Distribution parameters and thresholds for residual of different data channels

	Normal distribution						Logistic distribution					
Ch	μ	σ	AIC	T^c			μ	s	AIC	T^c		
				90%	95%	99%				90%	95%	99%
FR	0.0004	0.27	221.16	0.59	0.79	1.24	0.006	0.14	119.98	0.31	0.42	0.66
ET	0.006	0.14	-1157.9	0.3	0.4	0.63	0.02	0.069	-1318.041	0.151	0.203	0.32
ICP	0.003	0.08	-2333.67	0.17	0.23	0.35	0.001	0.041	-2396.17	0.09	0.12	0.2

These figures show that the proposed unhealthy detection strategy is able to detect the unhealthy states of FR, ET, and ICP data channels successfully. The total number of 54 unhealthy states have been detected by the logistic threshold during this test period, including 36 unhealthy states in FR data channel, 2 in ET data channel, and 16 in ICP data channel. As illustrated in Figure 2.10 on page 63, the unhealthy timestamps can be merged to determine the time intervals for further investigation in which the vehicle might not be operating correctly. For instance, analyzing the unhealthy timestamps in FR data channel results in determining five unhealthy time intervals for further investigation. Moreover, the unhealthy timestamps in ICP data channel can be summarized as four unhealthy intervals for further analysis. Moreover, we compared the recognized unhealthy states of these data channels with the vehicle maintenance indicator record to validate the efficiency of the proposed approach. The comparison results confirm the accuracy of the proposed approach in modeling the behavior of the operating vehicle and detecting the anomalous behaviors of these data channels. According to the maintenance record, the time period that an engine maintenance was conducted matches the unhealthy time intervals captured by the proposed approach.

Table 2.10

False positive and false negative rates of distributions in different data channels

Channel	Normal distribution						Logistic distribution					
	90%		95%		99%		90%		95%		99%	
	FP[a]	FN[b]	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN
FR	0.0	0.132	0.0	0.156	0.0	0.156	0.105	0.004	0.016	0.031	0.0	0.136
ET	0.004	0.061	0.0	0.061	0.0	0.061	0.038	0.053	0.008	0.061	0.0	0.061
ICP	0.0	0.039	0.0	0.050	0.0	0.062	0.108	0.008	0.012	0.012	0.0	0.042
Average	0.001	0.077	0.0	0.089	0.0	0.093	0.084	0.022	0.012	0.035	0.0	0.08

[a] Anomaly false positive rate
[b] Anomaly false negative rate

2.4.6 Reliability Analysis

In this section, the Receiver Operating Characteristic (ROC) curve is used to evaluate the reliability of the proposed hybrid model in detecting the unhealthy states of the operating vehicle. The ROC curve is one of the widely used tools for analyzing the quality and reliability of anomaly detection models. This probability curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various classifying thresholds. So, the bigger the TPR and the smaller the FPR are, the better the model is in detecting abnormalities. In other words, it determines how much the proposed model is capable of distinguishing between the healthy and unhealthy states of the vehicle. Hence, the performance of the hybrid ARIMA-WNN model for detecting the unhealthy states is compared with the WNN and ARIMA models on different data channels. Figure 2.11 on page 64 shows the ROC curves for FR and ICP data channels. For instance, for detecting the unhealthy states of FR channel, the areas under the ROC curve (AUC) of the ARIMA-WNN, WNN, and ARIMA models are 0.888, 0.767, and 0.741, respectively. Similarly, the AUC of ARIMA-WNN, WNN, and ARIMA models respectively are 0.977, 0.699, and 0.573 for ICP data channel. The results indicate that, comparing with the component models (i.e., ARIMA and WNN), the proposed hybrid model has better ability to understand the underlying characteristics of different time series data. Therefore, using the ARIMA-WNN improves the accuracy of the anomaly detection by increasing the true positive rate and decreasing the false positive rate.

2.5 Summary and limitation

This study presents a hybrid ARIMA-WNN model for predicting the behavior of the operating vehicle and detect the unhealthy states based on multiple channel time series data. The proposed hybrid method incorporates the ARIMA model in the first stage and the WNN model in the second stage for predicting the time series values. The ARIMA model constructs the linear component of prediction in the first stage and then the WNN model is programmed on the residual errors of the first stage model to generate the nonlinear component of prediction in the second stage. By integrating ARIMA and WNN, the proposed hybrid model exploits the ARIMA capability for predicting the unseen patterns in unlabeled historical data and WNN flexibility for analyzing time series data with different variations. A threshold based anomaly detection strategy is developed to timely recognize the unhealthy states of the vehicle. Fitting the maximum likelihood distribution for the residual errors of the hybrid model, the classifying thresholds of different data channels are computed. Once a residual error of a data channel exceeds the threshold, an unhealthy alert is issued. A feed-forward neural network and backpropagation method are nested into this hybrid model for training the WNN model. Since the WNN model is developed without any labeled data, it can work in an unsupervised setting. This model addresses the drawbacks of the classical neural network such as weight initialization, local minima, and model complexity.

Compared with ANN, the proposed model is more tolerant to noise and more sensitive to temporal anomalies. A large scale multiple time series vehicle operating data is used as a real case study to test the performance of the proposed hybrid ARIMA-WNN and compare with ARIMA, and WNN results. This data set includes three months of second-wise records for 101 performance data channels of a specific vehicle. Running the proposed ARIMA-WNN, WNN, and ARIMA models on different data channels reveals the higher accuracy of the hybrid model comparing with other models in modeling the time series behavior of the vehicle. For instance, compared with ARIMA, the ARIMA-WNN model improves the MAE, RMSE, and NSE for 58.19%, 66.42%, and 32.04% when running on the test data of the FR data channel. It also improves the WNN model by 30.21% in MAE, 34.3% in RMSE, and 0.55% in NSE indicators when predicting the test data of the ET data channel. Besides, in terms of RMSE, ARIMA-WNN model improves ARIMA and WNN models for 75.5% and 29.92% on the test data of the ICP data channel. Furthermore, the residual errors of the hybrid ARIMA-WNN model for predicting different data channels are evaluated to identify the unhealthy states of the operating vehicle. Evaluating different probability distributions (e.g., normal, logistic, gamma), the best fitted distributions for the residuals of different channels are determined to construct the respective classifying thresholds. Testing different residual probability distributions, the normal and logistic distributions are chosen due to their reasonable likelihood and AIC values. Additionally, among three different classifying confidence levels (i.e., 90%, 95%, and 99%), the classifying level 95% is chosen since it results in a reasonable trade-off between the false negative and false positive unhealthy rates. By estimating the proper distributions for the residuals of different channels, the classifying thresholds are then computed to efficiently identify the unhealthy states of the vehicle. Results confirm the proficiency of the proposed threshold based anomaly detection strategy for detecting the abrupt changes in the behavior of different data channels. Finally, the capability of the hybrid ARIMA-WNN and ARIMA models for distinguishing between the healthy and unhealthy states of different data channels is compared using ROC curves. For instance, the AUC value of the FR data channel is enhanced from 0.573 in the ARIMA model to 0.977 in the proposed hybrid model that indicates the ability of the ARIMA-WNN model in understanding the underlying time series process of this data channel.

The proposed framework also has managerial insight contributions. First of all, the ARIMA-WNN approach is able to model and predict vehicle behaviors which will help to better understand the vehicle's operating status and provide the appropriate recommendations for either predictive or preventive maintenance. Furthermore, the anomaly detection technique will support the unhealthy status identification, which would alert the potential harms to the managers so as to mitigate the operating risk and avoid the further economic loss. Additionally, the proposed framework incorporating the ARIMA-WNN model and anomaly detection strategy will provide a general time series analysis mechanism to be applied to the large scale multiple channel time series data analysis in many other domains, including marketing and stock pricing.

Although this study has several novel contributions to the state of the art, there are still limitations that require future research. First, the proposed hybrid ARIMA-WNN, ARIMA, and WNN models predict the time series of different data channels separately. Then, unhealthy states of the vehicle are identified based on these predictions for individual data channels. Hence, developing a multi-variate prediction model and anomaly detection strategy in the vehicle time series data is worthy of investigation. Furthermore, other modeling approaches such as Hidden Markov modeling strategies or Recurrent Neural Network models can be further studied. Finally, transfer learning can also be investigated to deal with the challenge of time-varying characteristics in the time series data of the operating vehicle. The future works will take into account all these limitations and extend the current research scope.

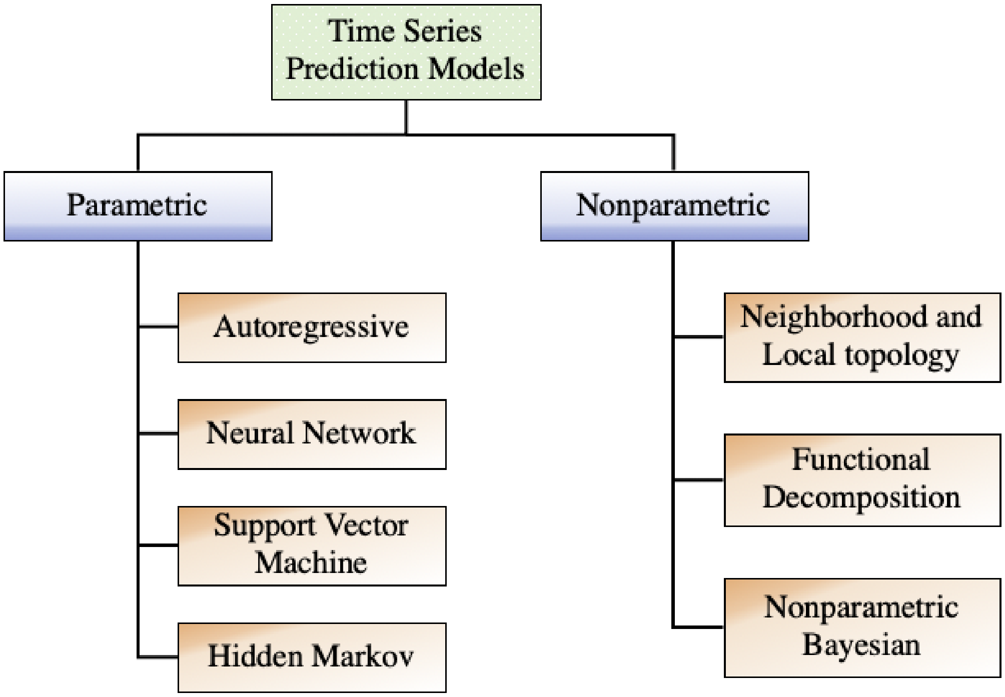


Figure 2.1

Classification of time series prediction models

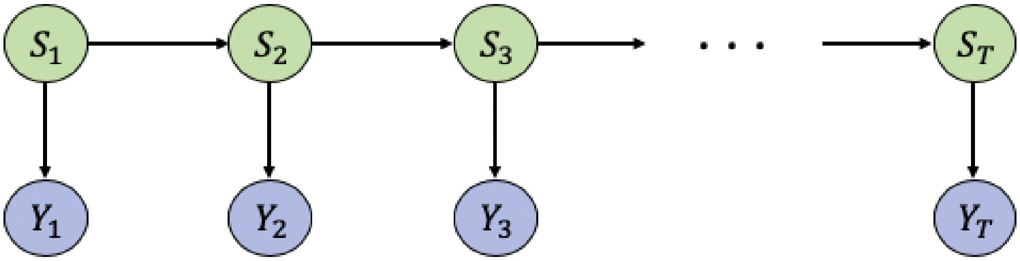


Figure 2.2

Graphical representation of an HMM

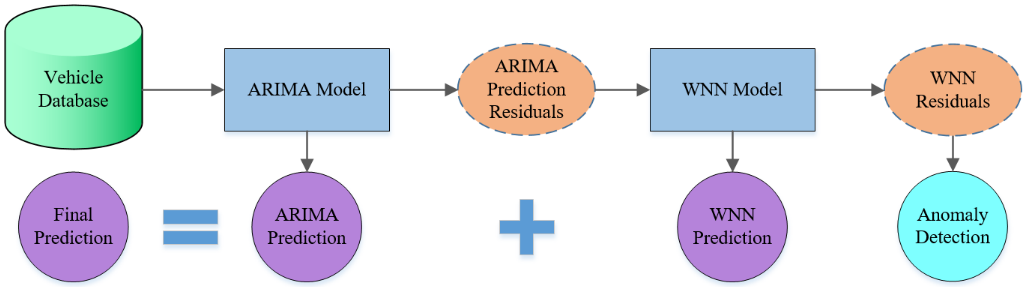


Figure 2.3

General structure of the hybrid ARIMA-WNN

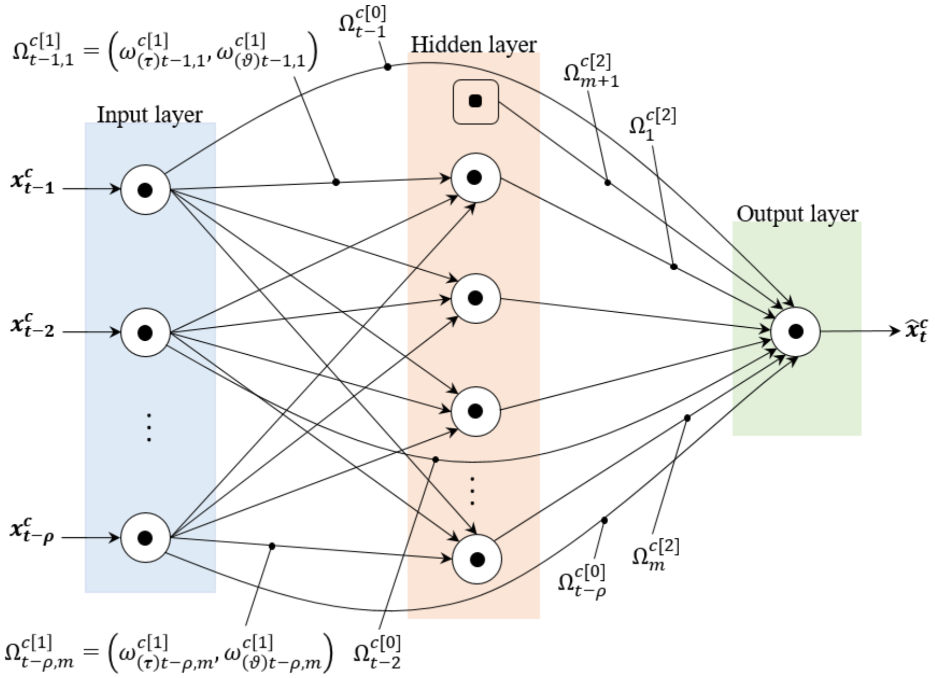
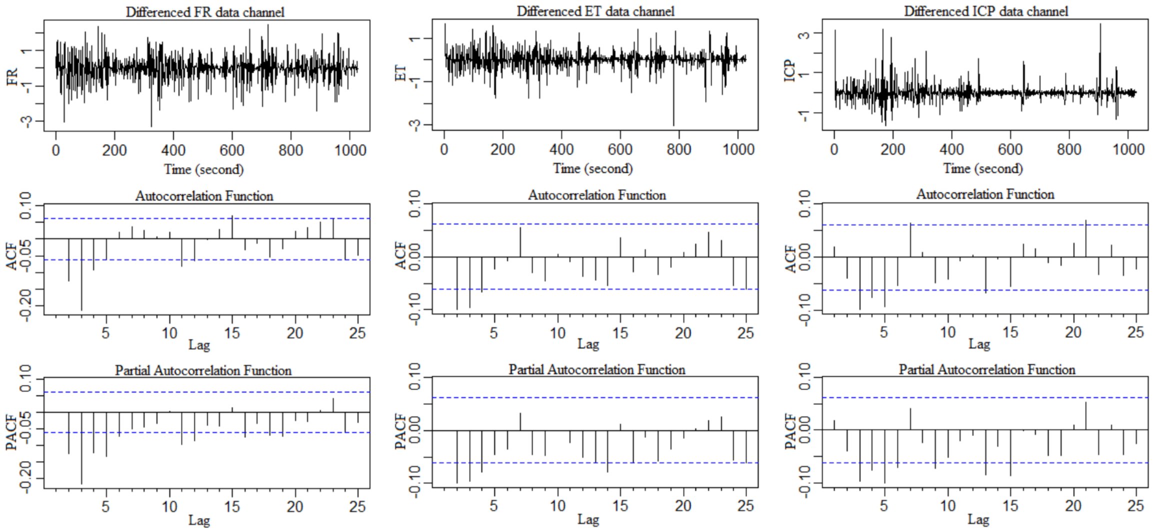

$\Omega^{c[1]}_{t-1,1} = \left(\omega^{c[1]}_{(\tau)t-1,1}, \omega^{c[1]}_{(\vartheta)t-1,1}\right)$
$\Omega^{c[0]}_{t-1}$
Hidden layer
$\Omega^{c[2]}_{m+1}$
Input layer
$\Omega^{c[2]}_{1}$
x^c_{t-1}
Output layer
x^c_{t-2}
$\hat{x}^c_t$
$x^c_{t-\rho}$
$\Omega^{c[2]}_{m}$
$\Omega^{c[0]}_{t-\rho}$
$\Omega^{c[1]}_{t-\rho,m} = \left(\omega^{c[1]}_{(\tau)t-\rho,m}, \omega^{c[1]}_{(\vartheta)t-\rho,m}\right)$
$\Omega^{c[0]}_{t-2}$

Figure 2.4

Architecture of the proposed WNN



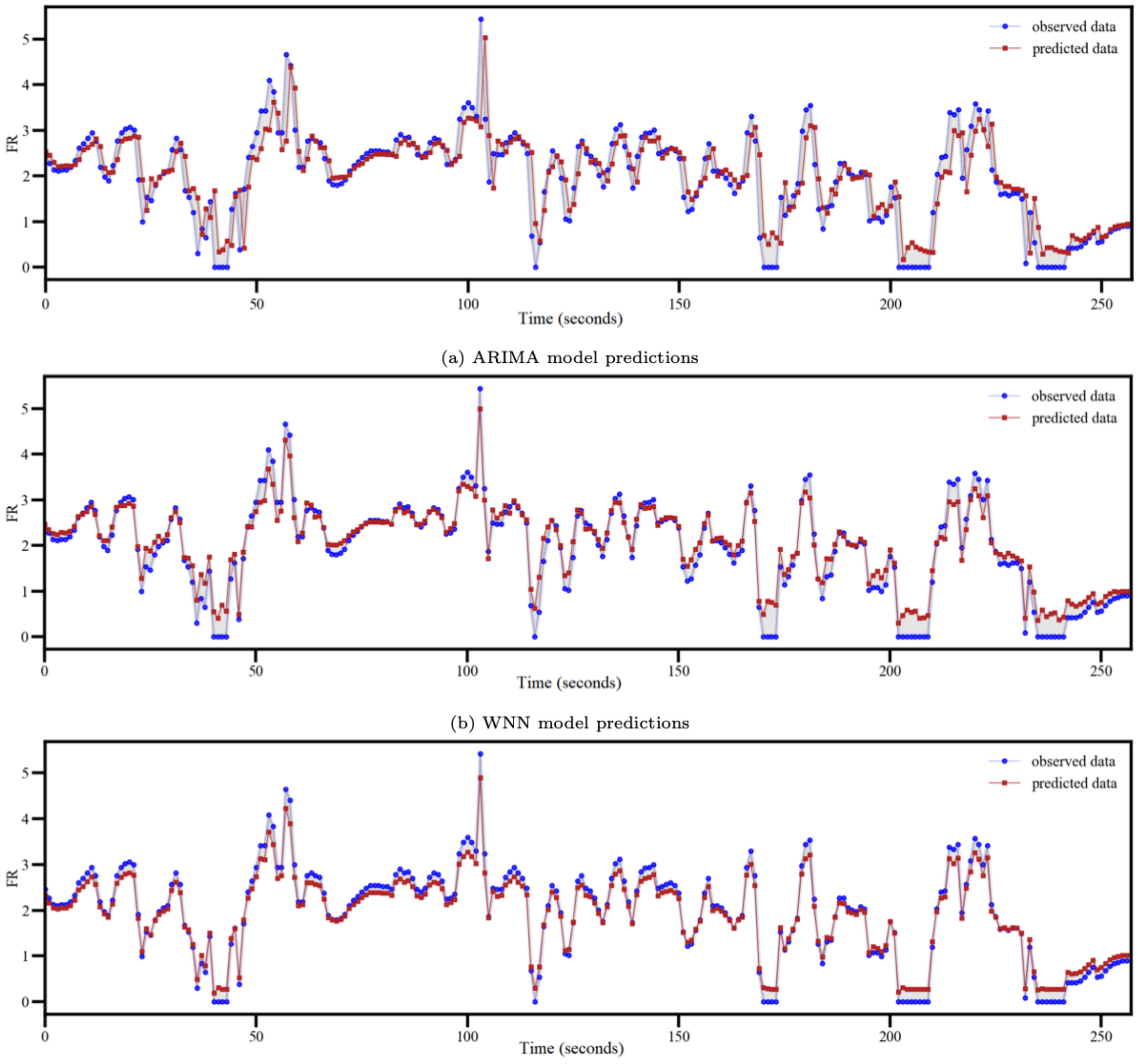
(a) Fuel rate data channel

(b) Engine torque data channel

(c) Injection control pressure data channel

Figure 2.5

Correlation analysis for determining the input lags for (a) FR, (b) ET, and (c) ICP data channels



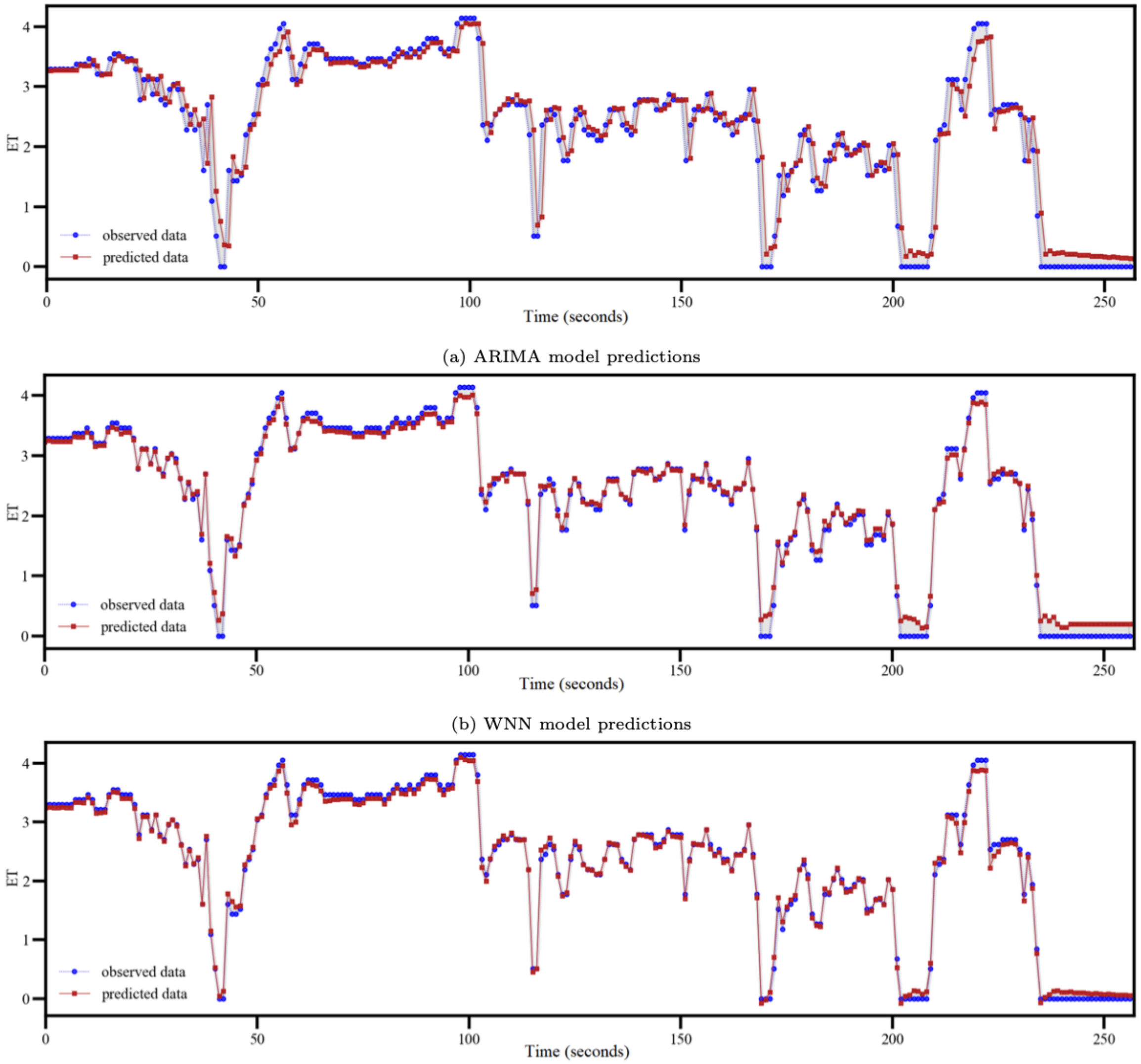
(a) ARIMA model predictions

(b) WNN model predictions

(c) ARIMA-WNN model predictions

Figure 2.6

Fuel rate data channel



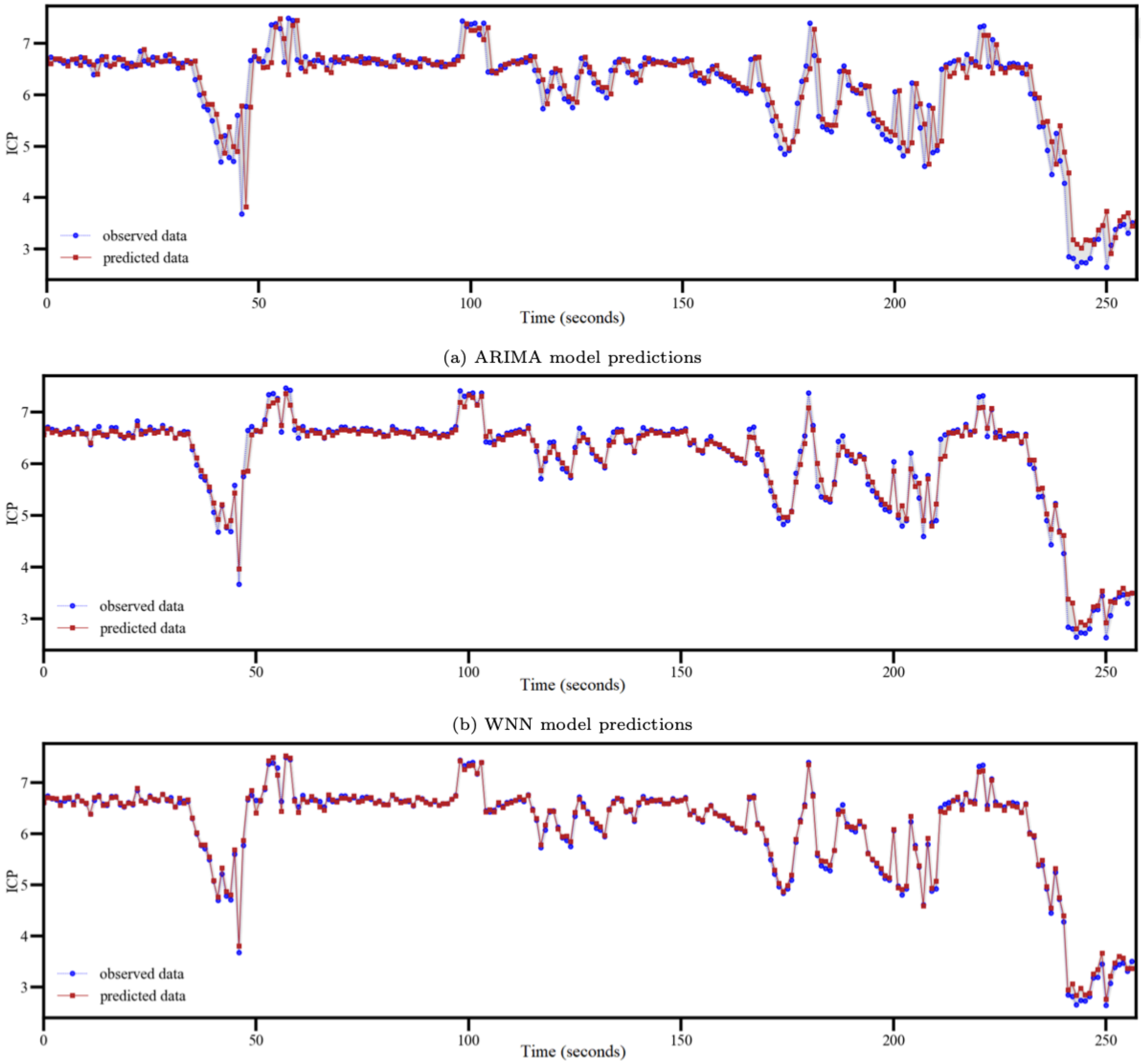
(a) ARIMA model predictions

(b) WNN model predictions

(c) ARIMA-WNN model predictions

Figure 2.7

Engine torque data channel



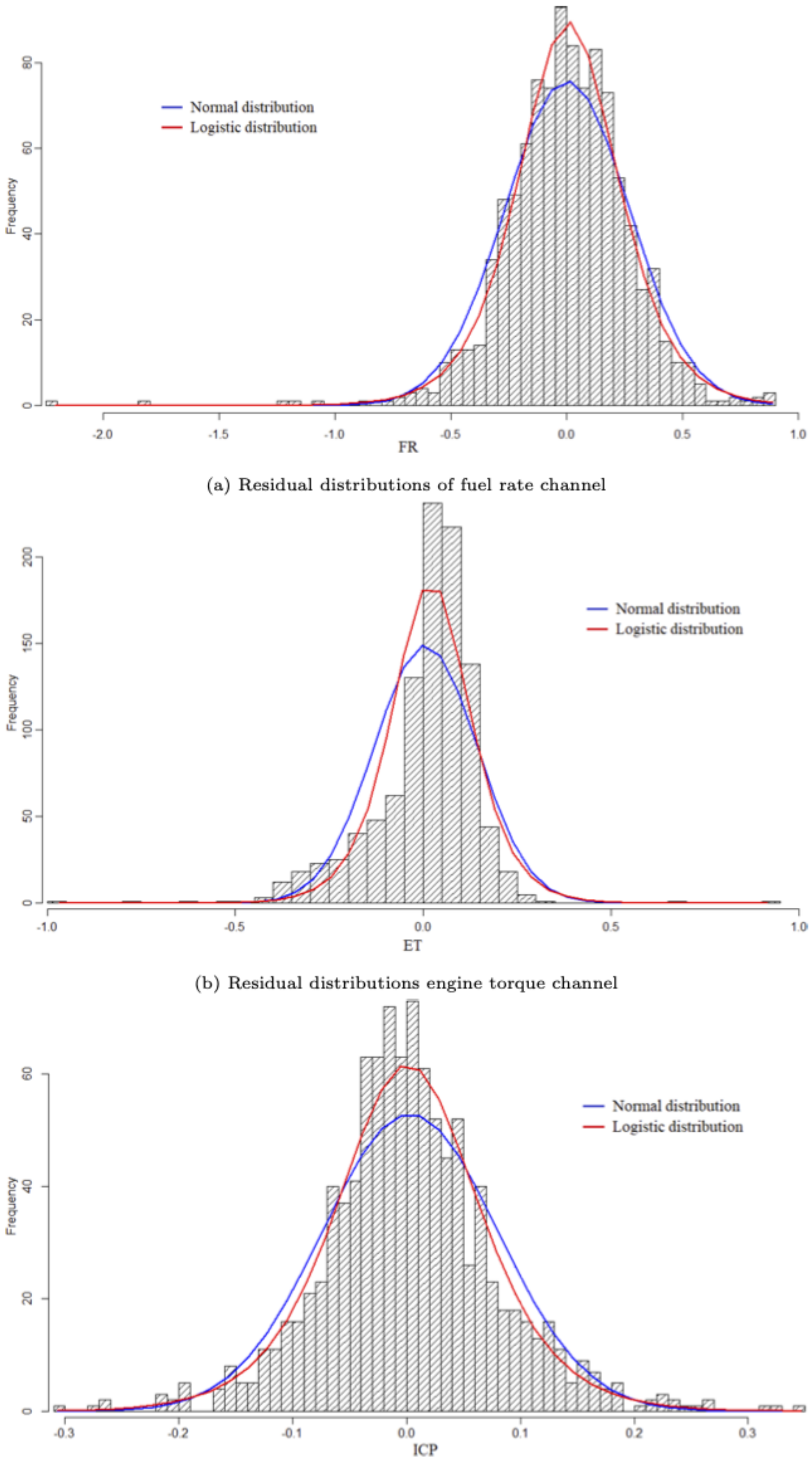
(a) ARIMA model predictions

(b) WNN model predictions

(c) ARIMA-WNN model predictions

Figure 2.8

Injection control pressure data channel



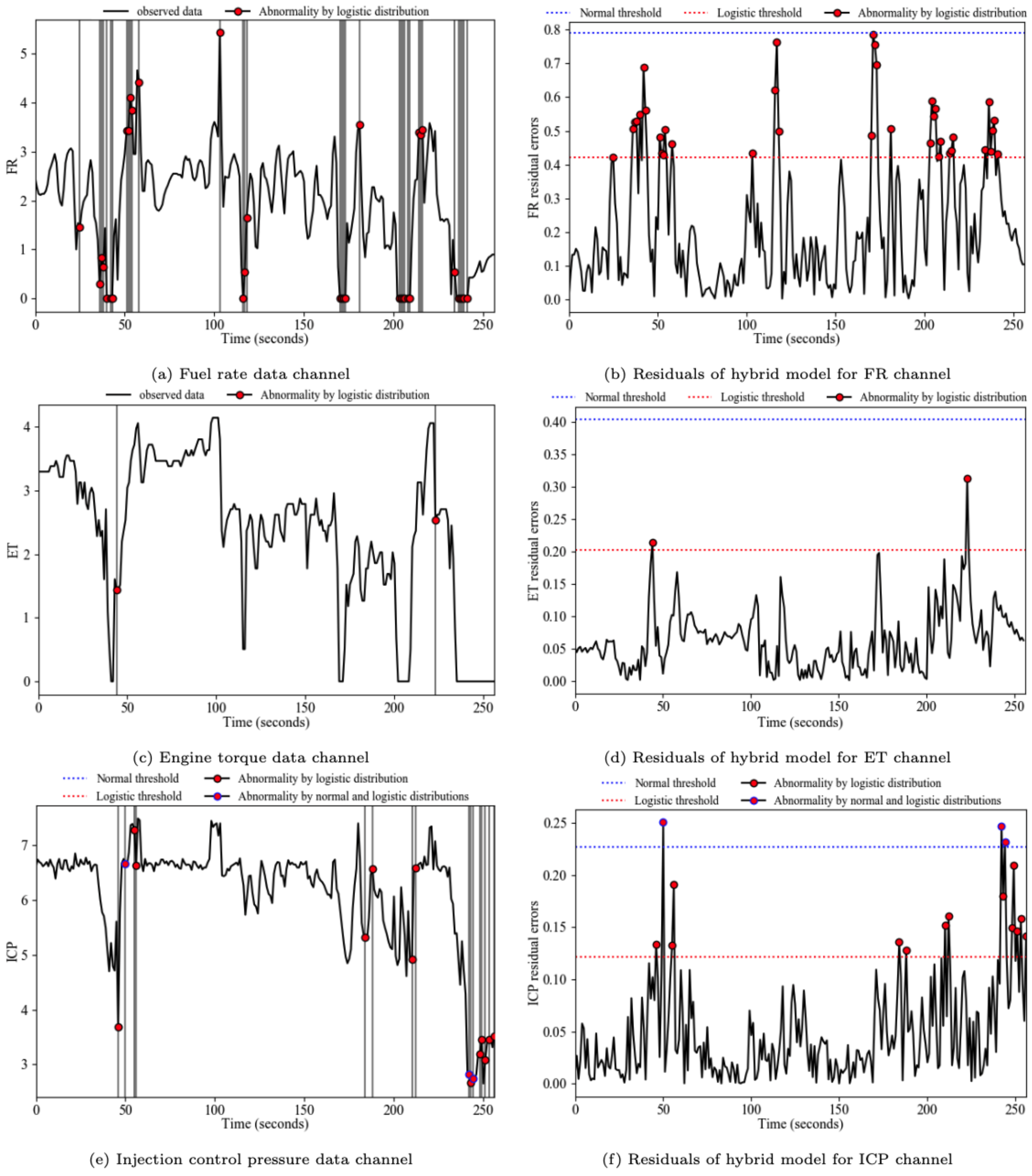
(a) Residual distributions of fuel rate channel

(b) Residual distributions engine torque channel

(c) Residual distributions of injection control pressure

Figure 2.9

Residual distributions for FR, ET, and ICP data channels



(a) Fuel rate data channel

(b) Residuals of hybrid model for FR channel

(c) Engine torque data channel

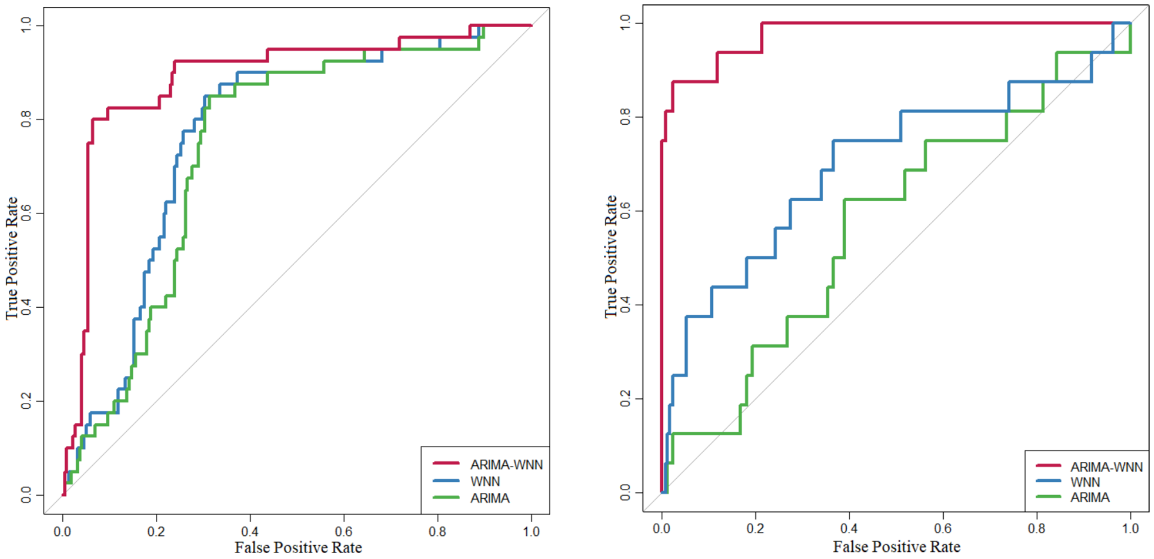
(d) Residuals of hybrid model for ET channel

(e) Injection control pressure data channel

(f) Residuals of hybrid model for ICP channel

Figure 2.10

Unhealthy states detection for FR, ET, and ICP data channels



(a) Fuel rate data channel

(b) Injection control pressure data channel

Figure 2.11

ROC curves for FR and ICP data channels

CHAPTER III

A COMPARATIVE STUDY OF SERIES HYBRID APPROACHES TO MODEL AND PREDICT THE VEHICLE OPERATING STATES

3.1 Introduction: Time series anomaly detection

Nowadays, the advancement of modern automobile technologies has motivated auto manufacturers to implement new technologies in the vehicle system to satisfy customers' diverse requirements. Utilizing multiple new technologies in the vehicle system have made it more complicated. Thus, monitoring the vehicle's behavior is vital to preserve the performance and help to extend the life cycle and decrease the operating and maintenance cost. The faulty performance of a vehicle's subsystem almost emerges a long time before being detected. In other words, an operating subsystem will behave differently once a fault begins to occur, which indicates the abnormal behavior of the vehicle that requires appropriate maintenance. Maintaining the nominal performance of an operating vehicle during the lifetime without monitoring its behavior is extremely challenging. One way to detect the imminent failures in the vehicle system is to model and predict the behavior of the subsystems and identify the abrupt variations. For this purpose, this paper investigates and compares various statistical analyzing and deep learning methods to model the behavior of operating subsystems, predict the forthcoming patterns, and identify the abnormal behaviors for multiple time series data.

Prediction models and anomaly detection have been widely studied across various domains such as manufacturing, lifeline systems, and telecommunication networks. Many prior studies have analyzed trends and anomalies using data-driven modeling approaches that can be generally categorized into two types: statistical methods and machine learning methods. In the statistical technique group, a statistical model is developed based on data assumed to be normally distributed. Then, anomaly scores can be computed based on the deviations or error residuals. The statistical models can be classified as parametric [208, 23, 4] and non-parametric [62, 128] models. However, these models can recognize unseen patterns and generate statistically significant solutions, they need a large amount of data and do not perform well when the predefined distribution is not appropriate. By training labeled data that are marked as normal or anomalous, machine learning techniques develop classification models to classify new data as anomalous or normal. There are various classification algorithms such as decision tree [21], support vector machine (SVM)[182], k-nearest neighbors [185], artificial neural network (ANN) [126], etc. These algorithms can classify unseen data accurately if there is an appropriate amount of labeled training data. Moreover, new unknown types of anomalies may be unfolded in the vehicle operating subsystems where no labeled training data existed. It would also be cumbersome for some classification algorithms to detect periodic or seasonal anomalies since they cannot recognize the temporal dependencies across timestamps.

It is widely accepted that combining different models or leveraging hybrid approaches can considerably improve the prediction accuracy and succeed in dealing with the limitations of single models [15]. This is due to the facts that the underlying process of real data generation cannot be easily determined or single models may not be able to appropriately identify the true data generation process. Furthermore, combining heterogeneous models or hybridization would result in lower generalization variance or error [48, 81]. In this paper, the performances of different series hybrid models for predicting the behavior of subsystems of an operating vehicle are evaluated. These models include ARIMA, MLPNN, WNN, ARIMA-MLPNN, ARIMA-WNN, MLPNN-ARIMA, and WNN-ARIMA. The main goal of this study is to evaluate the predictive capabilities of these models and investigate which sequence of models is better to construct series hybrid models for predicting subsystems behaviors. Furthermore, an effective anomaly detection method is proposed to identify the abnormal behaviors based on the predicting results of the best model. This approach detects abnormalities by thresholding the residual errors of prediction and normalizing the dependence on the magnitude of the prediction values.

3.2 Prior Studies

Time series data is a common data type that presents data over time. To involve time series characteristics in the modeling, many temporal approaches have been developed to detect anomalies based on timestamps predictions. These models usually don't need any primary data distribution and labeled training data. Autoregressive integrated moving average (ARIMA) method has been extensively used to model the behavior and detect abnormalities with consideration of temporal dependencies [130, 206, 232]. [225] proposed ARIMA models to detect platoon and mobility anomalies and design a two-step prediction model for diminishing the false alarms due to road curves. [148] studied traffic characterizations and abnormality detection in network management by applying the ARIMA model and traditional Holt-Winters methods. [146] developed an ARIMA-based anomaly detection model to specify the traffic network behavior and recognize the traffic anomalies. [209] introduced an improved ARIMA model to detect traffic abnormalities in wireless sensor networks. [187] developed an ARIMA-based anomaly detection method to monitor patients activities in several closed ward hospitals. [90] applied ARIMA technique to detect anomalies in the information system data collected through regular vehicle sensors to efficiently score and rank drivers. Although these models are sensitive to noise, they would high-likely to generate false positive anomaly flag when the noise level is severe [198].

To address this issue in time series analysis, some recent research studied wavelet reconstruction methods that combine wavelet basis functions (WBFs) and ANN in analyzing variations and making predictions. As a pre-processing step, these techniques decompose the original data to multiple scales and then different neural networks will be separately applied to analyze each component. For instance, [165] integrated wavelet transforms and auto-associative neural networks to treat transient signals for recognizing novelties or anomalies of faulty signals in rotating machinery. [190] combined discrete wavelet transform and ANN to diagnose high impedance faults. [89] proposed a signal processing algorithm by incorporating wavelets, ANN, and Hilbert transform to detect short and long-term anomaly patterns in time series data. [173] also combined wavelet transformation, ANN, and high-frequency surrogate measurements to detect water quality anomalies in water management systems. Wavelet reconstruction methods facilitate the training and anomaly detection in ANNs by decomposing the original time series data to several scales, however, these techniques may be difficult to perform online. Because, the ANN of different components are trained separately and then the obtained results will be combined, which will be labor-intensive and time-consuming. Some other research applied WBF as a transformation function in the hidden layer of ANN [224], which take the self-organizing advantage of ANN and time-frequency properties of WBF, and therefore outperform the conventional ANN [199, 153, 164, 180]. For instance, [6] presented a self-recurrent neural network relying on wavelets architecture with multidimensional radial wavelons for detecting network intrusions. [116] constructed a WNN model by using modified quantum-behaved particle swarm optimization (MQPSO) algorithm for network anomaly detection. [82] developed an anomaly detection model by combining normalized mutual information feature selection (NMIFS) and quantum WNN. [193] proposed a WNN model to detect cyber intrusions and anomalies for industrial control communication. [198] integrated WNN, classifying threshold, and two detecting strategies to identify anomalies in ocean fixed-point time series data. More recently, [210] presented a WNN model to predict the electrical load of midterm buildings.

In recent years, studies pertaining to combination techniques of prediction models have attracted a great deal of attention [70]. These combination models can be mainly classified into two categories: series and parallel models. Parallel combination models generate the hybrid predictions by the combination of predicting results of single models; while series combination models consider decomposing time series data into two main components and analyzing them separately. The first model analyzes one of the components of the time series in the first stage and then another component is modeled by the second stage model based on the results obtained from the first stage. Series combination models, especially linear/nonlinear combination, are among the most commonly used hybrid approaches for time series prediction [216]. Several literature studies constructed various ARIMA-ANN models by combining ARIMA with multilayer perceptrons neural network (MLPNN)[13, 196], Elman's recurrent neural networks [5], radial basis function neural network [171], and probabilistic neural networks [93] for time series prediction. Moreover, [40] integrated seasonal autoregressive integrated moving average (SARIMA) and SVM methods to predict seasonal time series. [140] studied a hybrid ARIMA-SVM model for predicting short-term loads in energy management systems. [53] combined ARIMA model with MLPNN and explanatory variables (ARIMAX) for air quality forecasting in urban areas. [54] studied the combination of ARIMA model with MLPNN and support vector regression (SVR) models and constructed two hybrid systems ARIMA-MLPNN and ARIMA-SVR for time series forecasting. Their proposed hybrid systems determine a suitable function to integrate the linear and nonlinear prediction components as well. Also, [48] proposed a particle swarm optimization (PSO) algorithm to find the best parameters of the linear and nonlinear components of a series ARIMA-ANN prediction model. Several studies combined ARIMA model with generalized auto-regressive conditional heteroskedasticity (GARCH) model for diagnosing machine health condition [150], short-time traffic flow prediction [38], solar radiation forecasting [178], and anomaly detection in network traffic [131, 11]. By combining ANN and GARCH models, [132] introduced a series hybrid model for predicting volatility in financial markets.

Some recent studies have also developed hybrid ARIMA-WNN models to improve prediction accuracy. Integrating multi-resolution analysis (MRA), WNN, and ARIMA, [81] proposed a multi-scale decomposition and reconstruction approach for predicting real-time traffic behaviors. [160] studied a hybrid ARIMA-WNN model to predict wheat yield time series data. [220] constructed a hybrid approach to predict short-term electricity loads in power systems by combining ARIMA, WNN, and improved empirical mode decomposition (IEMD). Later, [155] introduced a hybrid ARIMA-WNN model to evaluate the performance of a cloud environment and identify short-term performance anomalies. In addition to applying ARIMA-ANN models in numerous studies, ANN-ARIMA models are also studied. By combining grey relational artificial neural network (GRANN) and ARIMA, [162] developed the hybrid GRANN-ARIMA model for time series forecasting.

[212] developed a hybrid ANN-ARIMA model to forecast short-term traffic flow time series. [94] evaluated the efficiency of ARIMA-ANN and ANN-ARIMA models for stock price forecasting. To determine the best sequence of single models in constructing bi-component series hybrid models, [71] compared the performance of various models including ARIMA-SVM, ARIMA–MLPNN, SVM–ARIMA, and MLPNN–ARIMA in predicting time series data. Despite the broad study of ARIMA-WNN models in various studies, WNN-ARIMA models have been studied in relatively fewer papers. To accurately predict the urban traffic flow, [77] integrated WNN model with ARIMA model using a fuzzy method. [134] proposed a hybrid WNN-ARIMA model for predicting the index of stock market time series.

In this research, the proposed series hybrid methodology and structure of ARIMA-MLPNN, ARIMA-WNN, MLPNN-ARIMA, and WNN-ARIMA models are described. Based on the benchmark case data, the performances of the proposed models in predicting the behavior of the vehicle subsystems are analyzed and the obtained results are reported. Moreover, the abnormal behaviors of the subsystems achieved by the thresholding method are presented.

3.3 ARIMA, MLPNN, WNN Models and their Series Hybridization

Individual approaches for time series forecasting can be mainly categorized into statistical and intelligent models. The statistical models such as ARIMA predict the time series based on the past values and prediction errors of the time series components. Intelligent models such as MLPNN and WNN do not rely on the form of the relationships between the input and output data. In other words, these models predict the outputs by analyzing the features of the input data. The major advantage of these individual models is their capability for modeling linear and nonlinear relationships. ARIMA, MLPNN, and WNN models are the most widely used statistical and intelligent models for time series forecasting. Moreover, these models have been frequently used in the literature for constructing series hybrid prediction models.

3.3.1 ARIMA Model

ARIMA model is a statistical approach to predict the future values of time series based on historical observations and random errors. The ARIMA(p, d, q) model mainly comprises autoregressive (AR) and moving average (MA) components and can be formulated as follows:

$$\phi(B)(1 - B)^d (x_t^c - \mu) = \theta(B)\varepsilon_t \tag{3.1}$$

where, x_t^c is the actual value of subsystem $c \in C$ in time $t \in \mathcal{T}$ and the white noise $\varepsilon_t \approx iid(0,\sigma^2)$. $\phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i$ and $\theta(B) = 1 - \sum_{j=1}^{q} \theta_j B^j$ show the polynomial functions of the backshift operator B with degree p and q, respectively. Furthermore, $\phi_i, i = 1, 2, \ldots, p$, and $\theta_j, j = 1, 2, \ldots, q$ are the model parameters, integers p and q stand for the orders of the model, and d refers to as the order of differencing. In other words, orders p and q indicate the number of x_t^c lags and the number of lagged errors in the ARIMA model for predicting future values of subsystem c, respectively.

Using the Box and Jenkins [214] method, the procedure of ARIMA modeling consists of three main steps: model identification, parameter estimation, and diagnostic checking. The identification step specifies the number of AR (p) and MA terms (q) and the number of differencing operations (d). Box and Jenkins [214] proposed the AutoCorrelation function (ACF) and the Partial AutoCorrelation function (PACF) of the sample time series data to identify the orders of the ARIMA model. After specifying the ARIMA(p,d,q), the ordinary least squared (OLS) method is used to estimate the parameters which were identified in the previous step. Finally, the diagnosis step checks the adequacy of the constructed ARIMA model. This step determines whether or not an adequate ARIMA model is constructed to predict the time series data well, as another ARIMA model may exist with better modeling and prediction performance. Therefore, the final structure of the ARIMA model is selected based on the diagnostic statistics squared estimate of errors (SSE), Akaike information criterion (AIC), and prediction residual plot. These criteria are formulated in previous chapter. Note that if the selected model does not fit the sample time series data adequately, a new ARIMA model will be constructed, and the three described steps will be repeated until the best model structure is found.

3.3.2 MLPNN model

ANN models are among the most widely used intelligent models for predicting time series data. The popularity is firstly due to the considerable capability of extrapolating the underlying data generation without any assumption of the model form. Secondly, these models are powerful universal estimators that can approximate a wide variety of functions. There are various ANN models in the literature with different architecture. Single hidden layer feed-forward (also known as multilayer perceptron) neural network (MLPNN) is the most frequently used neural network design for modeling and predicting time series data. In this study, MLPNN is used for modeling the nonlinear relationships in the time series data. The proposed model consists of three layers including input, hidden, and output layers.

The input layer comprises the past lagged values of actual observations $x^c_{t-1}, x^c_{t-2}, \ldots, x^c_{t-\rho}$ for subsystem c of the vehicle. The ρ nodes of input layer are individually connected to all nodes of hidden layer. The hidden layer operates between the input and output layers with m nodes that are all connected to the single node of the output layer. This layer is operated by an activation function $\phi(\cdot)$ which defines the relationship between the input and output layers. Neural network models support a large class of activation functions such as linear, logistic, quadratic, and tanh. In this paper, the common logistic function is used as the transfer function in the hidden layer of the MLPNN model as follows:

$$\phi(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

By choosing the logistic activation function and the appropriate number of nodes m in the hidden layer, the single node of output layer can predict the future values of time series for different subsystems of the vehicle. Since the one-step-ahead forecasting approach is considered in this paper, the output layer only contains one node. Thus, the MLPNN model predicts t-th value of time series c as follows:

$$\hat{x}_t^c = \omega_{m+1}^{c[2]} + \sum_{j=1}^{m} \omega_j^{c[2]} \phi\Big(\omega_{\rho+1,j}^{c[1]} + \sum_{i=1}^{\rho} \omega_{i,j}^{c[1]} x_{t-i}^c\Big) \tag{3.3}$$

where $\omega_{i,j}^{c[1]}\ \forall\ i = 1, \ldots, \rho + 1,\ j = 1, \ldots, m + 1$ stands for the connection weights between the input and hidden layers and $\omega_j^{c[2]}\ \forall\ j = 1, \ldots, m + 1$ shows the connection weights between the hidden and output layer. To start the algorithm, these weights are randomly initialized in small values between 0 and 1. Also, $\hat{x}_t^c$ corresponds to the model prediction for t-th value of time series in subsystem c of the vehicle.

3.3.3 WNN Model

As a new class of ANN models, WNN models incorporate wavelet analysis in the hidden layer of the model to improve the time series predictions. [224] introduced WNN to deal with some drawbacks of regular ANN models such as random weight initialization, local minima, and model complexity. Similar to the MLPNN model, the proposed WNN model consists of an input layer with ρ nodes, a hidden layer with m nodes, and an output layer with one node. The input layer comprises the lagged observations which are connected to the hidden layer and the single node of the output layer. The hidden layer contains the hidden nodes (also known as wavelons) which transfer the input variables to translated and dilated versions of the mother wavelet. Finally, the output layer predicts the future values for the time series of different subsystems. The details of the proposed WNN models are presented in Chapter 2.

3.3.4 The series hybridization of ARIMA, MLPNN, and WNN Models

This subsection presents the proposed series combination of the individual models. In series linear/nonlinear hybrid models, the time series data of each subsystem $c \in C$ is considered to comprise a linear autocorrelation and a nonlinear component as follows:

$$x_t^c = \Upsilon_t^c + \Phi_t^c \tag{3.4}$$

where Υ_t^c and Φ_t^c respectively show the linear and nonlinear components that are estimated from the time series data of c-th subsystem of the vehicle. All proposed hybrid models (i.e., ARIMA-MLPNN, ARIMA-WNN, MLPNN-ARIMA, and WNN-ARIMA) consider that the linear part is estimated by the ARIMA model whereas the nonlinear part is computed by either the MLPNN model or the WNN model. For instance, in the ARIMA-MLPNN model, the linear component of time series is modeled by ARIMA in the first stage and then the nonlinear component is modeled by MLPNN in the second stage using residuals of the first stage. Likewise, in the WNN-ARIMA model, the nonlinear part is estimated by WNN in the first stage and the linear part is modeled by

ARIMA in the second stage using residuals of the WNN model. The main idea of series hybrid model is that if the time series of a subsystem is modeled by a linear model such as ARIMA, the residuals of prediction only contain nonlinear relationships. Alternatively, if the time series is modeled by a nonlinear model such as MLPNN or WNN, the residuals only comprise the linear structure. Therefore, series combinations of individual models ARIMA, MLPNN, and WNN exploit the unique attitudes and strengths of these models for determining different patterns. It could also be beneficial to model the linear and nonlinear components separately and then combine the predictions to improve the overall forecasting performance. Because, in real applications such as vehicle operating data, it is difficult to completely understand the underlying characteristics of data. Hence, modeling different components of the time series of a subsystem sequentially by using the individual linear or nonlinear model could enhance the prediction accuracy. The below descriptions present different hybrid models constructed by series combination of ARIMA, MLPNN, and WNN models.

3.3.4.1 The ARIMA-MLPNN and ARIMA-WNN models

According to the series hybridization approach, in both ARIMA-MLPNN and ARIMA-WNN models, the ARIMA model constructs the linear part of time series prediction in the first stage. Let r_t^c represent the residual of the ARIMA model for subsystem c at time t as follows:

$$r_t^c = x_t^c - \hat{\Upsilon}_t^c \tag{3.5}$$

where $\hat{\Upsilon}_t^c$ denotes the linear output of the ARIMA model. The ARIMA model leaves the nonlinear patterns in the residual of the first stage. Then, in the second stage, the MLPNN explores the nonlinear relationships in the ARIMA residual using the ARIMA-MLPNN hybrid model.

Analogously, using the ARIMA-WNN model, the WNN model discovers the nonlinear component of the ARIMA residual in the second stage. With ρ nodes in the input layer, the WNN model or MLPNN model for the residual of the ARIMA model can be presented as follows:

$$\hat{\Phi}_t^c = f(r_{t-1}^c, r_{t-2}^c, \ldots, r_{t-\rho}^c) \tag{3.6}$$

$$r_t^c = \hat{\Phi}_t^c + \varepsilon_t^c \tag{3.7}$$

where f denotes the nonlinear function determined by either the WNN model or the MLPNN model. Additionally, $\hat{\Phi}_t^c$, and ε_t^c indicate the nonlinear prediction value and the corresponding random error. Note that if the nonlinear function f is inappropriate, the error term ε_t^c will not be necessarily random. Consequently, the combined prediction of sample t in subsystem c is computed as follows:

$$\hat{x}_t^c = \hat{\Upsilon}_t^c + \hat{\Phi}_t^c \tag{3.8}$$

Moreover, Figure 3.1 on page 95 represents the general frameworks of the ARIMA-MLPNN and ARIMA-WNN models in Figure 3.1 on page 95(a) and the MLPNN-ARIMA and WNN-ARIMA models in Figure 3.1 on page 95(b).

3.3.4.2 The MLPNN-ARIMA and WNN-ARIMA models

Similarly, the MLPNN-ARIMA and WNN-ARIMA models have two main parts. The only difference is that in these models, the nonlinear part of time series is first constructed. Then, the second stage model determines the linear part based on the residual of the first stage. Thus, in

MLPNN-ARIMA or WNN-ARIMA, the nonlinear term $\hat{\Phi}_t^{c'}$ is initially estimated by MLPNN or WNN. Let $r_t^{c'}$ denote the residual of these nonlinear models as follows:

$$r_t^{c'} = x_t^c - \hat{\Phi}_t^{c'} \tag{3.9}$$

Then, in the second stage, the ARIMA model is fitted to the residual of nonlinear models to estimate the linear term $\hat{\Upsilon}_t^{c'}$ as follows:

$$\hat{\Upsilon}_t^{c'} = f(r_{t-1}^c, r_{t-2}^c, ..., r_{t-\rho'}^c) \tag{3.10}$$

$$r_t^{c'} = \hat{\Upsilon}_t^{c'} + \varepsilon_t^{c'} \tag{3.11}$$

where f is a linear function specified by the ARIMA model, $\hat{\Upsilon}^{c'}$ denotes the linear prediction part generated by the ARIMA model, and ε_t^c is the random error. Thus, the MLPNN-ARIMA or WNN-ARIMA make the final prediction by combining the nonlinear and linear components as follows:

$$\hat{x}_t^{c'} = \hat{\Phi}_t^{c'} + \hat{\Upsilon}_t^{c'} \tag{3.12}$$

3.3.4.3 Abnormal Behavior Detection Method

Evaluating different individual (i.e., ARIMA, MLPNN, WNN) and hybrid models (ARIMA-MLPNN, ARIMA-WNN, MLPNN-ARIMA, and WNN-ARIMA), a model with the best performance is considered to model the normal behavior and detect the abnormal behavior of different subsystems of the vehicle. That is, a significant deviation from the behavior prediction of the best model is considered as abnormal behavior. Given the prediction $\hat{x}_t^c$ and the actual observation x_t^c, the prediction error $\varepsilon_t^c = \hat{x}_t^c - x_t^c$ is used as a deviation metric to identify the abnormal behavior of different subsystems. If the absolute value of error falls outside the pre-defined threshold, an abnormal behavior alert is issued. Using hypothesis testing, the maximum likelihood distribution of residual errors ε_t^c is determined. The prediction error sequence follows a certain distribution (e.g., normal, logistic, lognormal, gamma, etc.). Hence, its probability distribution can be estimated by using the maximum likelihood estimation approach. Thus, the classifying threshold T^c for detecting the abnormal behavior of subsystem c can be defined [198] as follows:

$$T^c = \frac{1}{2}\left[\left|a + \ln(\frac{p_{up}}{1 - p_{up}})b\right| + \left|a + \ln(\frac{p_{down}}{1 - p_{down}})b\right|\right] \quad \forall\, c \in C \tag{3.13}$$

where a and b are the parameters of the probability distribution fitted to the residuals of different subsystems. For normal distribution, $a = \mu$ and $b = \sigma$ are mean and standard deviation. For logistic distribution, $a = \mu$ and $b = s$ are the location and scale parameters. Moreover, p_{up} denotes the classifying level (e.g., 90%, 0.95%, 99%, etc.) and p_{up}+p_{down}=1. Since the proposed threshold is computed based on the classifying level instead of user-defined value, this method constructs the threshold in the form of predictive levels that requires no experiential knowledge or parameter measurements.

3.4 Results and discussion

In this section, the results obtained by running different individual and hybrid models on the time series of FR, ET, and ICP subsystems of the vehicle are presented.

3.4.1 Individual ARIMA Model For Different Subsystems

The stationarized series, ACF, and PACF of FR, ET, and ICP subsystems are presented in Figure 3.2 on page 96. The blue lines in these ACF and PACF plots show the lower and upper bounds of the confidence interval. It means that correlation coefficients falling within the confidence

Table 3.1

Models evaluation for FR, ET, and ICP data channels

Channel	Best model	AIC	SSE
FR	ARIMA(4,1,2)	1848.81	357.97
ET	ARIMA(3,1,4)	1018.94	159.56
ICP	ARIMA(5,1,5)	901.24	141.32

bounds will result in no significant lag whereas falling outside of the confidence interval indicates a significant lag. Figure 3.2 on page 96(a) shows that the ACF outside the confidence interval occurs at lags 2 to 4, and lags 11, 12, and 15 for the FR subsystem. So the t-2, t-3, and t-4 as well as t-11, t-12, and t-15 are chosen as the relevant lags of prediction errors. Moreover, as the PACF falls outside of the confidence bands at lags 2 to 6 and lags 11, 12, 16, 18, and 19, the t-2 to t-6 in addition to the t-11, t-12, t-16, t-18, and t-19 are selected as the relevant lags of previous observations. Analogously, for the ET subsystem in Figure 3.2 on page 96(b), the main relevant lags of x_t^c are t-2, t-3, and t-4 and the lags of prediction errors are t-2, t-3, t-4, and t-14. Also, according to Figure 3.2 on page 96(c) the previous observations lags and the prediction errors lags respectively are t-3 to t-5, t-7, t-13, and t-21 as well as t-3 to t-5, t-6, t-9, t-13, and t-15 for the ICP subsystem. Evaluating various combinations of the order parameters, Table 3.1 represents the best-fitted individual ARIMA models for different subsystems. In this table, the best ARIMA models are selected based on low AIC and SSE values. The lower value of these metrics indicates a better fit for the ARIMA model. In other words, low AIC and SSE values show quality models with less complexity and lower training error.

Table 3.2

Individual and hybrid models configuration for different subsystems

Sub-system	MLPNN				WNN				ARIMA-MLPNN				ARIMA-WNN				MLPNN-ARIMA				WNN-ARIMA			
	ρ	m	γ	η	ρ	m	γ	η	ρ	m	γ	η	ρ	m	γ	η	ρ	m	γ	η	ρ	m	γ	η
FR	8	16	0.05	-	3	6	10^{-4}	10^{-1}	4	9	10^{-3}	-	8	14	10^{-2}	10^{-1}	8	16	0.05	-	3	6	10^{-4}	10^{-1}
ET	8	19	10^{-2}	-	8	10	10^{-4}	10^{-1}	4	9	10^{-4}	-	3	6	10^{-2}	10^{-4}	8	19	10^{-2}	-	3	6	10^{-2}	10^{-4}
ICP	5	14	10^{-2}	-	4	8	10^{-3}	0.05	4	12	10^{-5}	-	6	5	10^{-2}	0.05	5	14	10^{-2}	-	4	8	10^{-3}	0.05

3.4.2 Hybrid ARIMA-MLPNN and ARIMA-WNN Models

Using the best-fitted ARIMA model, the linear components of these hybrid models are constructed in the first stage. Then, the ARIMA-MLPNN/ARIMA-WNN applies the best-fitted MLPNN/WNN model in the second stage for modeling the nonlinear relationships in the residual of the previous step. The MLPNN and WNN model are trained for different dimensions of the input vector in the input layer $\rho \in [1,25]$, the various number of neurons in the hidden layer $m \in [2, 10 \times \rho]$, and one output neuron. Among different structures of the WNN model, the best model will be obtained with the minimum training error. Table 3.2 indicates the optimal values of ρ, m, γ, and η parameters to construct the best MLPNN, WNN, and hybrid models for different subsystems. It should be noted that the MLPNN and WNN models are trained by the time series of different subsystems as individual prediction models, and in the second stage to evaluate the performance of the proposed hybrid models.

3.4.2.1 Fuel Rate Subsystem

For the FR subsystem, the best-fitted model ARIMA(4,1,2) estimates the linear component in the first stage. According to the ARIMA-MLPNN parameters for FR subsystem in Table 3.2, the best MLPNN model in this hybrid approach consists of 4 input, 9 hidden, and one output neurons. Similarly, the best WNN parameters in the ARIMA-WNN model for fitting the FR time series include 8 input, 14 hidden, and one output nodes. Finally, the obtained results from the first and the second stages of each hybrid model are combined to estimate the prediction of the hybrid approach for the FR time series. Figure 3.3 on page 97 represents the estimated values of the individual ARIMA, MLPNN, and WNN models against the actual test data of FR time series. Furthermore, Figure 3.4 on page 98 shows the predictions of the hybrid models ARIMA-MLPNN and ARIMA-WNN for the test data of FR subsystem.

3.4.2.2 Engine Torque Subsystem

As results in Table 3.1 on page 80 shows, the best-fitted linear model for ET time series is ARIMA(3,1,4). Analyzing the residual of ARIMA model by ARIMA-MLPNN, the best MLPNN construction comprises 4 input, 9 hidden, and one output nodes. Analogously, the best WNN structure for ET time series in ARIMA-WNN includes 3 input, 6 hidden, and one output nodes. The estimated values of the individual ARIMA and WNN models and the predictions of hybrid ARIMA-MLPNN and ARIMA-WNN models by combining the linear and nonlinear components against the actual values of ET test data are plotted in Figure 3.5 on page 99 and Figure 3.6 on page 100, respectively.

3.4.2.3 Injection Control Pressure Subsystem

Similar to the previous subsystem, the best ARIMA model fitting the time series of ICP subsystem is ARIMA(5,1,5). Then, the best nonlinear model for the residual of ARIMA model consists of 4 input, 12 hidden, and one output nodes in ARIMA-MLPNN and 6 input, 5 hidden, and one output nodes in ARIMA-WNN models. Figure 3.7 on page 101 shows the prediction

Table 3.3

Second stage ARIMA configuration for FR, ET, and ICP subsystems

Sub-system	MLPNN-ARIMA			WNN-ARIMA		
	Best model	AIC	SSE	Best model	AIC	SSE
FR	ARIMA(11,0,4)	334.33	80.11	ARIMA(8,0,6)	-1294.72	16.23
ET	ARIMA(3,0,6)	-2128.29	7.29	ARIMA(10,0,9)	-2974.17	3.10
ICP	ARIMA(9,0,5)	-1292.09	16.35	ARIMA(7,0,9)	-477.35	36.03

of the individual ARIMA and WNN models for the ICP subsystem. Integrating the linear and nonlinear parts, the estimated values of hybrid ARIMA-MLPNN and ARIMA-WNN models for this subsystem are presented in Figure 3.8 on page 102.

3.4.3 Hybrid MLPNN-ARIMA and WNN-ARIMA models

In these hybrid models, the MLPNN/WNN model is initially fitted to the time series of different subsystems for capturing the nonlinear patterns in the first stage. Then, the residual errors of the first stage are treated by the ARIMA model in the second stage for gathering the linear relationships. Finally, the estimations of the first and the second stage models are combined to construct the final prediction of the proposed MLPNN-ARIMA and WNN-ARIMA models.

3.4.3.1 Fuel Rate Subsystem

As shown in Table 3.2 on page 81, the MLPNN with 8 input, 16 hidden, and one output nodes has the best configuration in the first stage of the MLPNN-ARIMA model to accurately estimate the nonlinear component of the FR series. Table 3.3 represents the order of the ARIMA structure as the second stage of the MLPNN-ARIMA and WNN-ARIMA models. Thus, ARIMA(11,0,4) has the best structure to estimate the linear component of the FR series based on the residual errors of the first stage model. Similarly, the best configuration of the first and the second stage models in WNN-ARIMA approach consists of 3 input, 6 hidden, and one output nodes for the WNN model and ARIMA(8,0,6) for the ARIMA model, respectively. Figure 3.9 on page 103 presents the estimated values of the MLPNN-ARIMA and WNN-ARIMA models against the actual test observations of FR subsystem.

3.4.3.2 Engine Torque Subsystem

For accurately estimating the ET series, Table 3.2 on page 81 indicates that the MLPNN-ARIMA approach incorporates 8 input, 19 hidden, and one output nodes in the first stage MLPNN model and an ARIMA(3,0,6) in the second stage model. Likewise, the WNN-ARIMA approach comprises 3 input, 6 hidden, and one output node in the first stage WNN model and an ARIMA(10,0,9) in the second stage model. Figure 3.10 on page 104 depicts the estimated values of these hybrid models against the test data of ET subsystem.

3.4.3.3 Injection Control Pressure Subsystem

According to Table 3.2 on page 81, the best MLPNN-ARIMA model for estimating the time series of ICP subsystem is decomposed of 5 input, 14 hidden, and one out nodes in the first stage MLPNN model and an ARIMA(9,0,5) in the second stage model. Similarly, the best WNN-ARIMA involves 4 input, 8 hidden, and one output nodes in the first stage WNN and an ARIMA(7,0,9) in the second stage models. Figure 3.11 on page 105 represents the prediction of these hybrid models for the test observations of the ICP subsystem.

3.4.4 Comparison of Hybrid Models Results

This subsection evaluates the predictive capability of the proposed hybrid models as well as the component models ARIMA, MLPNN, and WNN. The performance indicators mean absolute error (MAE), root mean square error (RMSE), and Nash–Sutcliffe model efficiency coefficient (NSE) were used to compare the forecasting efficiency of the hybrid models and their components.

Table 3.4 on page 106 represents the training and testing errors of various models for different subsystems FR, ET, and ICP. However, the WNN model improves the prediction accuracy over different subsystems comparing to the ARIMA and MLPNN models, it has lower forecasting efficiency than the hybrid approaches. Table 3.5 on page 107, Table 3.6 on page 107, and Table 3.7 on page 108 represent the improvement percentages of hybrid models comparing to the component models for different subsystems FR, ET, and ICP, respectively. Numerical results show that the hybrid models outperform their component models. This may suggest that none of the ARIMA, MLPNN, and WNN models can completely capture the underlying trend of data in different subsystems. For instance, in terms of MAE, the ARIMA-MLPNN and MLPNN-ARIMA models respectively improve the prediction for FR test data by 58.3% and 69% comparing to the ARIMA model and by 43.2% and 57.7% comparing to the MLPNN model. From the RMSE perspective, the ARIMA-WNN and WNN-ARIMA models respectively enhance the FR test series prediction by 66.4% and 85.7% comparing to the ARIMA model and by 33.3% and 71.5% comparing to the WNN model. For ET test data, the ARIMA-MLPNN and MLPNN-ARIMA models can respectively improve the MAE of the ARIMA model by 64.5% and 73.5% and the MLPNN model by 38.9% and 54.4%. Likewise, the ARIMA-WNN and WNN-ARIMA models can respectively enhance the RMSE of predicting the ET test series by 80.4% and 85.6%

comparing to the ARIMA model and by 34.3% and 51.7% comparing to the WNN model. For ICP subsystem, the ARIMA-MLPNN and MLPNN-ARIMA models improve the MAE of ARIMA Model for 71.5% and 63.5%, respectively. These hybrid models can also improve the MAE of MLPNN model for 41.3% and 60.6%, respectively. Similarly, the ARIMA-WNN model reduces the RMSE of the ARIMA and WNN models for predicting the ICP test data by 75.5% and 29.9%, respectively. Moreover, the WNN-ARIMA model respectively reduces the RSME of the ARIMA and WNN models for predicting the ICP test data by 85.6% and 51.7%. Table 3.8 on page 108 shows the average improvements of different models over these three subsystems. It can be seen that the ARIMA-MLPNN and MLPNN-ARIMA models respectively enhance the average NSE of the ARIMA model for 19.3% and 19.9% and the MLPNN model for 4% and 4.5%. Besides, the ARIMA-WNN and WNN-ARIMA models respectively increase the average NSE of the ARIMA model by 19.9% and 21.2% and the WNN model by 1.8% and 2.9%. Improving the average NSE reveals the higher predictive performance of the hybrid models comparing to the component models. The average training errors over different subsystems are also decreased by the hybrid models comparing to the component models. The obtained results verify that the proposed hybrid models exploit the capacity of the ARIMA model in modeling the linear relationships and the capability of the MLPNN and WNN models in capturing the nonlinear patterns. Therefore, using hybrid models could be advantageous to separately determine the linear and nonlinear components by using different models and then combine these components to make the final prediction, which improve the overall predicting performance in different subsystems.

Furthermore, changing the sequence of using component models ARIMA, MLPNN, and WNN in the hybrid approaches highly impacts the overall prediction performance. Note that, in linear-nonlinear combination (i.e., ARIMA-MLPNN and ARIMA-WNN models), the ARIMA model is fitted in the first stage to capture the linear relationships and the MLPNN/WNN model determines the nonlinear patterns in the second stage. Conversely, in nonlinear-linear combination (i.e., MLPNN-ARIMA and WNN-ARIMA models), the MLPNN/WNN model captures the nonlinear relationships in the first stage and the ARIMA model determines the linear patterns in the second stage. For instance, in terms of MAE, the MLPNN-ARIMA model improves the ARIMA-MLPNN by 25.6% and ARIMA-WNN by 25.8% for predicting the test series of FR subsystem. The WNN-ARIMA model respectively reduces the RMSE of ARIMA-MLPNN, ARIMA-WNN, and MLPNN-ARIMA by 61.6%, 57.3%, and 47.1% for predicting the FR test data. For predicting the ET test data, the ARIMA-WNN and MLPNN-ARIMA models respectively enhance the ARIMA-MLPNN by 33% and 16.2% in terms of RMSE. Analogously, the WNN-ARIMA model improves the MAE of ARIMA-MLPNN, ARIMA-WNN, and MLPNN-ARIMA by 53.4%, 35.4%, and 37.7 % for predicting the ET test series, respectively. For predicting the ICP test data, the ARIMA-WNN model reduces the MAE of the ARIMA-MLPNN and MLPNN-ARIMA by 6.1% and 26.4%, respectively. Also, in terms of RMSE, the WNN-ARIMA model respectively improves the other hybrid models ARIMA-MLPNN, ARIMA-WNN, and MLPNN-ARIMA by 32.7%, 32.5%, and 51.7% for predicting the ICP test series. Therefore, as Table 3.8 on page 108 shows, the nonlinear-linear combination models WNN-ARIMA and MLPNN-ARIMA can yield slightly better performance than linear-nonlinear models ARIMA-WNN and ARIMA-MLPNN on average. For instance, the MLPNN-ARIMA model respectively decreases the MAE of ARIMA-MLPNN and ARIMA-WNN models by 13.3 % and 4.7% on average. Likewise, the WNN-ARIMA model respectively increases the NSE of ARIMA-MLPNN and ARIMA-WNN models by 1.6% and 1.1%

on average, which indicates its higher predictive skills. Moreover, comparing the two nonlinear-linear hybrid models for predicting various subsystems, the WNN-ARIMA performs better than the MLPNN-ARIMA on average, with 46.9%, 47.3%, and 1.1% improvements in MAE, RMSE, and NSE indicators, respectively.

3.4.5 Abnormal Behavior Detection

The WNN-ARIMA model showed better performance comparing to the other hybrid and component models for predicting the series of different subsystems. Thus, the residual errors of the predictions generated by this model are analyzed to detect the abnormal behavior of the operating vehicle based on different subsystems. Testing various probability distributions, the normal and logistic distributions are selected to fit the residual errors of the WNN-ARIMA model for different subsystems. Table 3.9 on page 109 shows the parameters, log-likelihood value (LLV), and AIC indicator of these distributions and the corresponding anomaly detection thresholds with different classifying levels for subsystems FR, ET, and ICP. The maximum likelihood estimation method is used to evaluate the fitness of these distributions for different subsystems.

As Table 3.9 on page 109 indicated, the logistic distribution with higher LLV and lower AIC values has a better fit to the residuals of various subsystems comparing to the normal distribution. For fitting the probability distribution to the residual errors of the FR subsystem, the LLV value increases from 667.78 for normal distribution to 729.39 for logistic distribution. Likewise, the AIC value decreases from -1331.55 to -1454.78 when respectively fitting the normal and logistic distributions to the residual errors of the FR subsystem. Comparing to the normal distribution, fitting the logistic distribution to the residual errors of the ET subsystem improves the LLV and AIC values for 3.22% and 3.23%, respectively. Also, fitting the logistic distribution to the residuals of the ICP subsystem increases the LLV and AIC values of the normal distribution by 47.94% and 48.11%, respectively. Hence, the threshold constructed based on the logistic distribution can be more effective for recognizing the abnormal behavior of the operating vehicle. Table 3.10 on page 110 shows the false positive (FP) and false negative (FN) rates of these probability distributions when the classifying level (CL) varies between 90% and 99%.

The FP and FN rates in this table indicate that increasing the classifying level from 90% to 99% increases the false negative rate and decreases the false positive rate for detecting abnormal behavior in different subsystems. In real applications like operating vehicle time series data, correctly identifying the actual abnormal observations with residual errors larger than the threshold is more important than those with residual errors smaller than the threshold. In other words, flagging the abnormal behaviors correctly is more significant than alarming the normal behaviors incorrectly. Testing different classifying levels 90%, 95%, and 99%, the reasonable level is chosen to hold a good trade-off between the false negative and false positive rates for detecting the abnormal behaviors in different subsystems. The obtained results represent that the threshold constructed based on the residual errors of the prediction model can reasonably determine the classifying boundary between the normal and abnormal behaviors of the vehicle. For FR subsystem, the logistic distribution has a location parameter of μ=0.0029 and a scale parameter of s=0.065, and the normal distribution has a mean value μ=-0.0012 and a standard deviation σ=0.126. Then, using equation 3.13, the corresponding thresholds are computed as $T^{FR}_{logistic} = 0.143$ and $T^{FR}_{normal} = 0.277$ by taking the classifying level 90%. For the ET subsystem with the classifying level 95%, the logistic threshold $T^{ET}_{logistic}$=0.085 and the normal threshold T^{ET}_{normal}=0.162 are computed based on the logistic distribution with μ=0.0009 and s=0.029 and the normal distribution with μ=-0.0002 and σ=0.055, respectively. Analogously, for the ICP subsystem with the classifying level 95%, the logistic threshold $T^{ICP}_{logistic}$=0.165 and the normal threshold T^{ICP}_{normal}=0.417 are generated by the logistic distribution with μ=0.0051 and s=0.056 and the normal distribution with μ=0.0011 and σ=0.142, respectively. Figure 3.12 on page 111 shows the normal and logistic probability distributions fitting to the residual errors of the FR, ET, and ICP subsystems. It can be seen that the logistic distributions are narrower and more centered on the mean than the normal distributions. Therefore, the classifying threshold constructed based on the logistic distribution is suggested in the presence of a small persistent sequence of abnormal behaviors while the normal distribution threshold is more applicable when the goal is to detect long-time anomalies.

Figure 3.13 on page 112 shows the abnormal behaviors that are identified by the logistic and normal thresholds. In this figure, the vertical grey lines show the actual abnormalities and the blue dots show the abnormalities detected by the logistic threshold. Besides, the blue dots with red edge color represent the anomalies identified by both normal and logistic thresholds. Clearly, the logistic threshold is more effective to identify the abnormal behaviors of the vehicle that are sudden persistent changes in the operating status of different subsystems. As shown in Figure 3.13 on page 112(a), the logistic threshold can detect the sudden changes in the fuel rate of the operating vehicle with lower FN rate 0.137 and higher FP rate 0.023 than the normal threshold with FN rate 0.186 and FP rate 0.0. Note that detecting the actual abnormality correctly is more significant than alarming the normal behaviors incorrectly. Thus, for better detecting the actual abnormalities in FR subsystem behavior, the logistic threshold is preferred due to its lower FN rate than the normal threshold. These abnormalities are detected based on the residual errors of WNN-ARIMA model that are higher than the logistic threshold in Figure 3.13 on page 112(b). Figure 3.13 on page 112(c) indicates that the abnormal behaviors of the ET subsystem are detected by the logistic threshold with lower FN rate 0.004 than the normal threshold with FN rate 0.05. Similarly, Figure 3.13 on page 112(e) presents the abnormal states in the ICP subsystem using logistic threshold since it has lower FN rate 0.023 comparing to the normal threshold with FN rate 0.035.

These abnormality plots verify the capability of the proposed anomaly detection method in detecting the abnormal behaviors of the operating vehicle based on the FR, ET, and ICP subsystems. The total number of 76 abnormal states have been detected by the logistic threshold during the given time period, including 21 unhealthy states in FR subsystem, 35 in ET subsystem, and 20 in ICP subsystem. As shown in Figure 3.13 on page 112, the abnormal timestamps of different subsystems can be merged to construct the time intervals for further investigation in which the vehicle might not be operating correctly. For example, the abnormal timestamps in FR subsystem can be summarized as five abnormal time intervals for further analysis. Similarly, the abnormal timestamps in ICP subsystem can be merged into four abnormal intervals for further investigation. Moreover, comparing the identified abnormal states of these subsystems with the vehicle maintenance record confirmed the effectiveness of the proposed approach for detecting the abnormal behaviors of the vehicle. According to the vehicle maintenance record, the time period that engine maintenance was conducted matches the abnormal time intervals captured by the proposed approach.

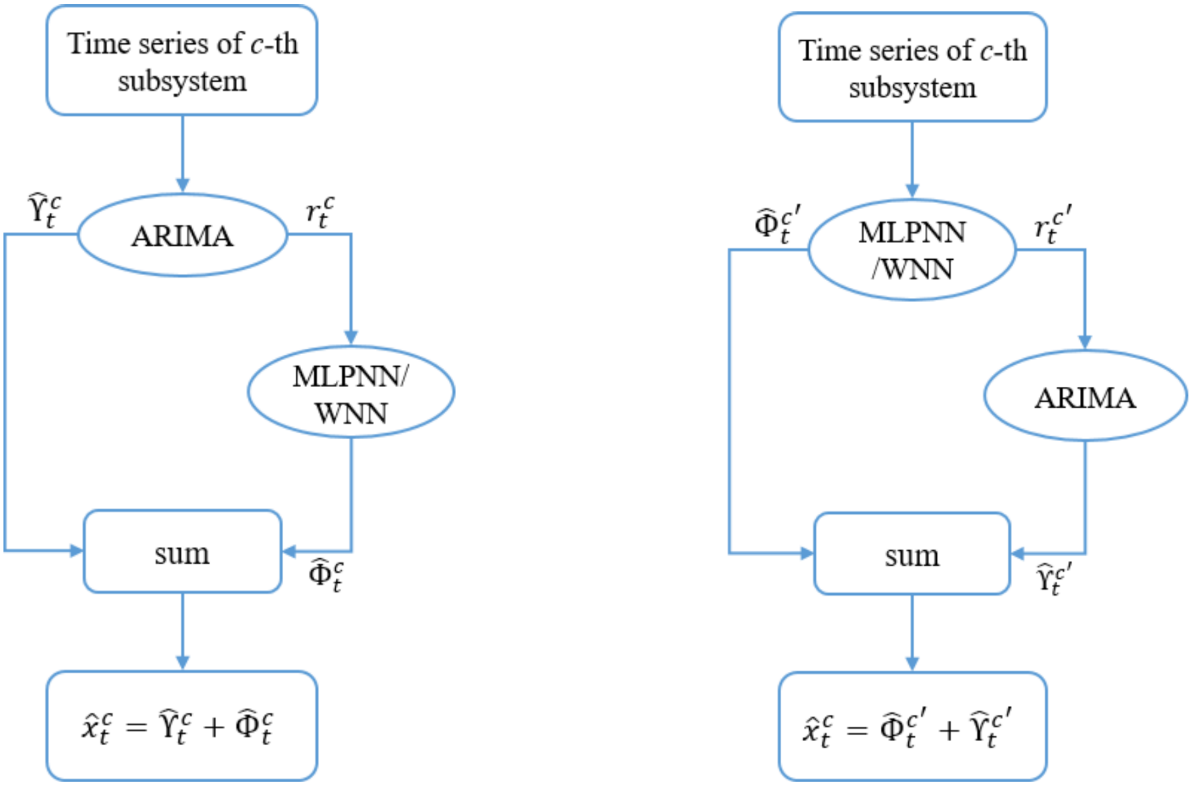
3.5 Summary and limitation

Nowadays, the growing complexity of modern vehicles requires decision-makers to monitor the behavior of the vehicle subsystems for preserving proper operating performance. This will not only extend the life cycle and decrease the maintenance costs of the vehicle, but also help with early detecting the imminent abnormal behaviors of operating subsystems. Making appropriate maintenance decisions for preserving the operating performance of the vehicle in its life cycle is highly challenging without appropriate modeling, predicting, and analyzing the time series records. By applying various statistical and deep learning techniques, this study develops several series hybrid models to model the behavior and predict the forthcoming pattern of multiple operating subsystems of the vehicle. Combining different predictive models is one of the efficient and most popular methods in the literature for overcoming the deficiency of single models and improving the prediction accuracy. Series hybrid models that decompose the time series data to linear and nonlinear components are among the widely-used hybrid models. These models exploit the unique advantages of linear and nonlinear models for capturing different relationships in the time series data.

This study investigates the predictive capability of four series hybrid linear/nonlinear combinations of autoregressive integrated moving average (ARIMA), multilayer perceptrons neural network (MLPNN), and wavelet neural network (WNN) models. The performance of these hybrid models (ARIMA-MLPNN, ARIMA-WNN, MLPNNN-ARIMA, and WNN-ARIMA) are compared mutually and with their component models for modeling the behavior and predicting the future trend of different subsystems. Moreover, a threshold-based anomaly detection method based on the best-fitted model is proposed to identify the abnormal behaviors of operating subsystems. Results with three months second-wise time series data of 101 subsystems in a real application reveal the higher accuracy of these series hybrid models comparing to the component models. For instance, the ARIMA-MLPNN and MLPNN-ARIMA models respectively reduce the prediction MAE for FR test data by 58.3% and 69% comparing to the ARIMA model and by 43.2% and 57.7% comparing to the MLPNN model. Analogously, in terms of RMSE, the ARIMA-WNN and WNN-ARIMA models improve the prediction of ICP test data by 75.5% and 85.6% comparing to the ARIMA model and by 29.9% and 51.7% comparing to the WNN model. Moreover, changing the sequence of using component models in the hybrid approaches highly impacts the overall prediction performance. For predicting the FR test data, the MLPNN-ARIMA model decreases the MAE of ARIMA-MLPNN and ARIMA-WNN models by 25.6% and 25.8%, respectively. Similarly, the WNN-ARIMA model respectively improves the RMSE of ARIMA-MLPNN, ARIMA-WNN, and MLPNN-ARIMA models by 50.8%, 26.5%, and 41.2% for predicting the test data of ET subsystem. The comparison of the four series hybrid models shows that the WNN-ARIMA model overall outperforms the other hybrid models. This model respectively enhances the NSE of ARIMA-MLPNN, ARIMA-WNN, and MLPNN-ARIMA models by 1.6%, 1.1%, and 1.1% which indicates its higher predictive skills. The obtained results demonstrate that the WNN-ARIMA model can be considered as an appropriate alternative for behavior forecasting in various operating subsystems of the vehicle. Furthermore, the proposed threshold-based anomaly detection method helps decision-makers to analyze the prediction residual errors and early detect the abnormal behaviors in different subsystems.

The findings of this study have significant contributions to vehicle system management. First and foremost, by comparing four series hybrid models, the most fitted model will be determined and this model will be helpful to understand the behavior of each vehicle subsystem over time. Furthermore, based upon the most fitted model results, the anomaly detection approach would detect abnormalities for the vehicle in the operation which would provide reliable and intelligent decision support for preventive and predictive maintenance for the vehicle and ultimately extend the vehicle life cycle. In addition, the vehicle behavior modeling and anomaly detection would provide recommendations and insights to vehicle system reliability analysis, and therefore support the vehicle system design so as to improve the reliability performance over the lifetime.

Although several novel contributions have made in this study, some limitations are needed to be tackled in future research. Since the proposed hybrid and component models analyze the time series of different subsystems separately, developing multi-variate prediction models for jointly analyzing various subsystems are worthy of investigation. Moreover, some other modeling approaches such as recurrent neural networks and hidden Markov models can be further studied. Finally, time series decomposition-based methods can also be used to develop another class of anomaly detection techniques for identifying abrupt changes in the behavior of the subsystems. The future study will extend the scope of this research to address all these limitations.

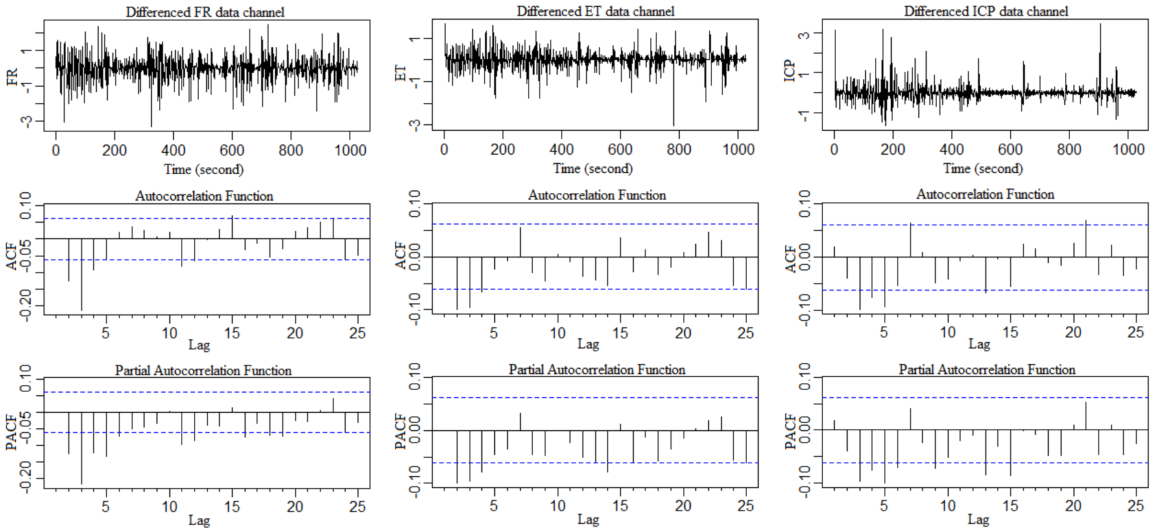

Time series of c-th subsystem
ARIMA
$\widehat{\Upsilon}_t^c$
r_t^c
MLPNN/ WNN
sum
$\widehat{\Phi}_t^c$
$\hat{x}_t^c = \widehat{\Upsilon}_t^c + \widehat{\Phi}_t^c$
Time series of c-th subsystem
MLPNN /WNN
$\widehat{\Phi}_t^{c'}$
$r_t^{c'}$
ARIMA
sum
$\widehat{\Upsilon}_t^{c'}$
$\hat{x}_t^c = \widehat{\Phi}_t^{c'} + \widehat{\Upsilon}_t^{c'}$

(a) ARIMA-MLPNN and ARIMA-WNN framework

(b) MLPNN-ARIMA and WNN-ARIMA framework

Figure 3.1

Frameworks of hybrid models



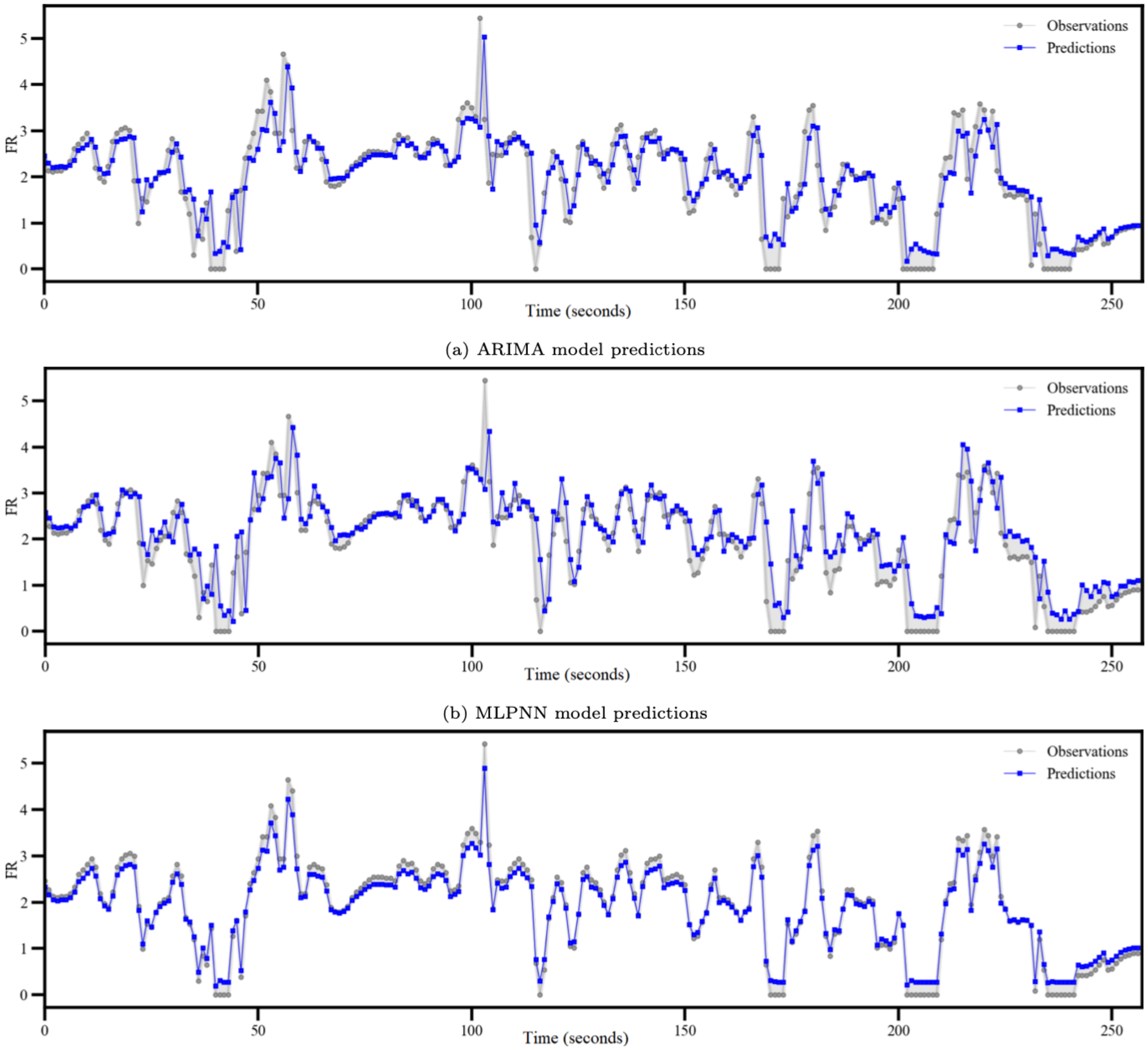
(a) Fuel rate data channel

(b) Engine torque data channel

(c) Injection control pressure data channel

Figure 3.2

Correlation analysis for determining the input lags for (a) FR, (b) ET, and (c) ICP data channels



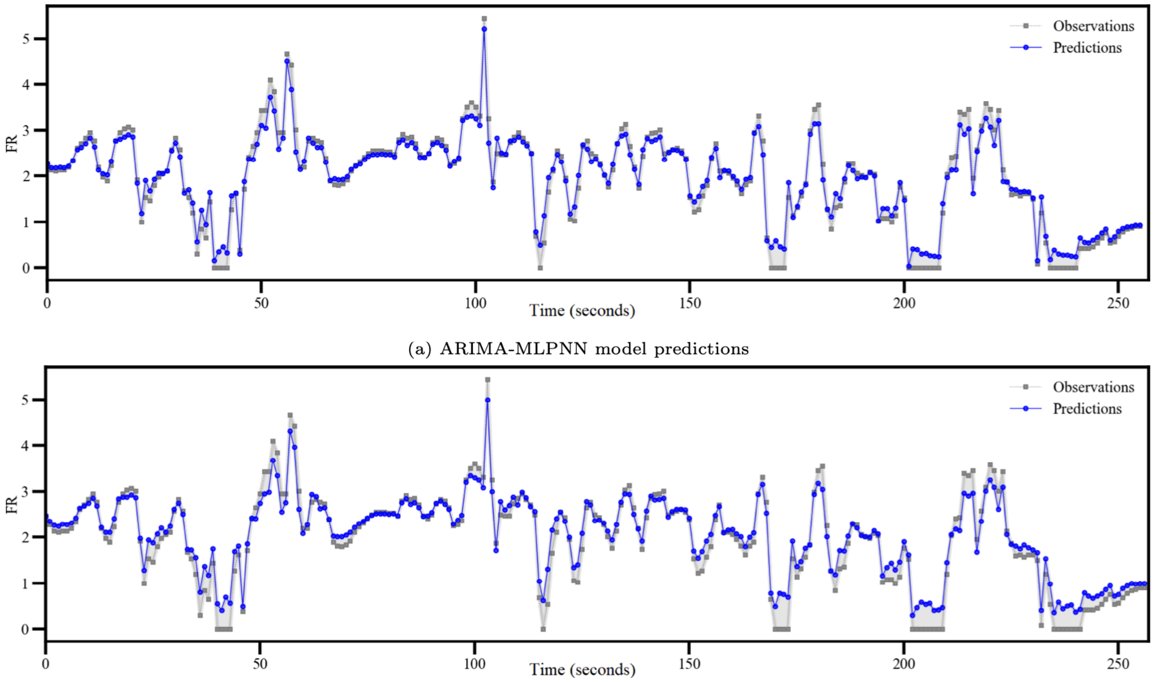
(a) ARIMA model predictions

(b) MLPNN model predictions

(c) WNN model predictions

Figure 3.3

Individual models predictions for FR subsystem

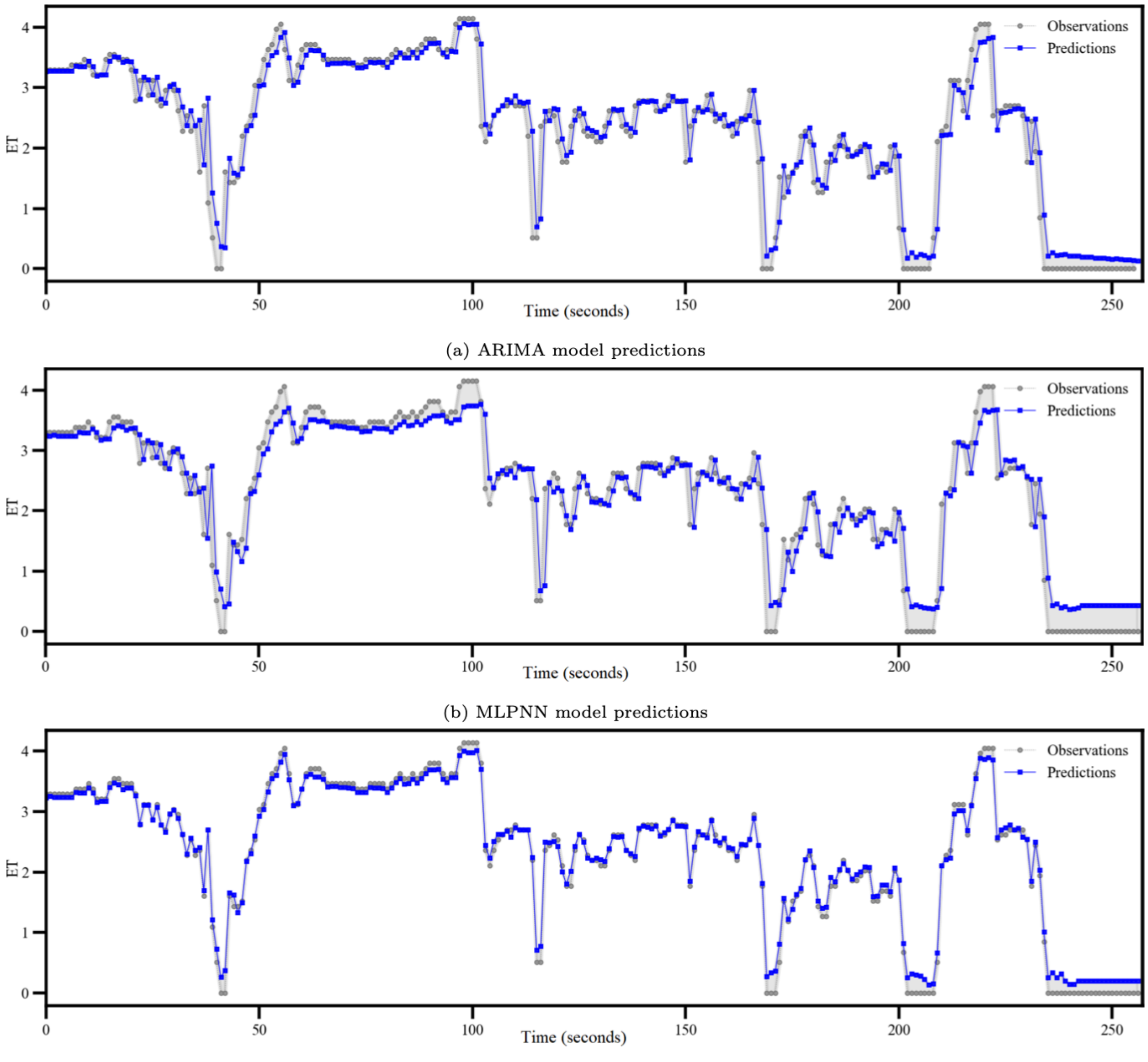


(a) ARIMA-MLPNN model predictions

(b) ARIMA-WNN model predictions

Figure 3.4

Hybrid ARIMA-MLPNN and ARIMA-WNN predictions for FR subsystem



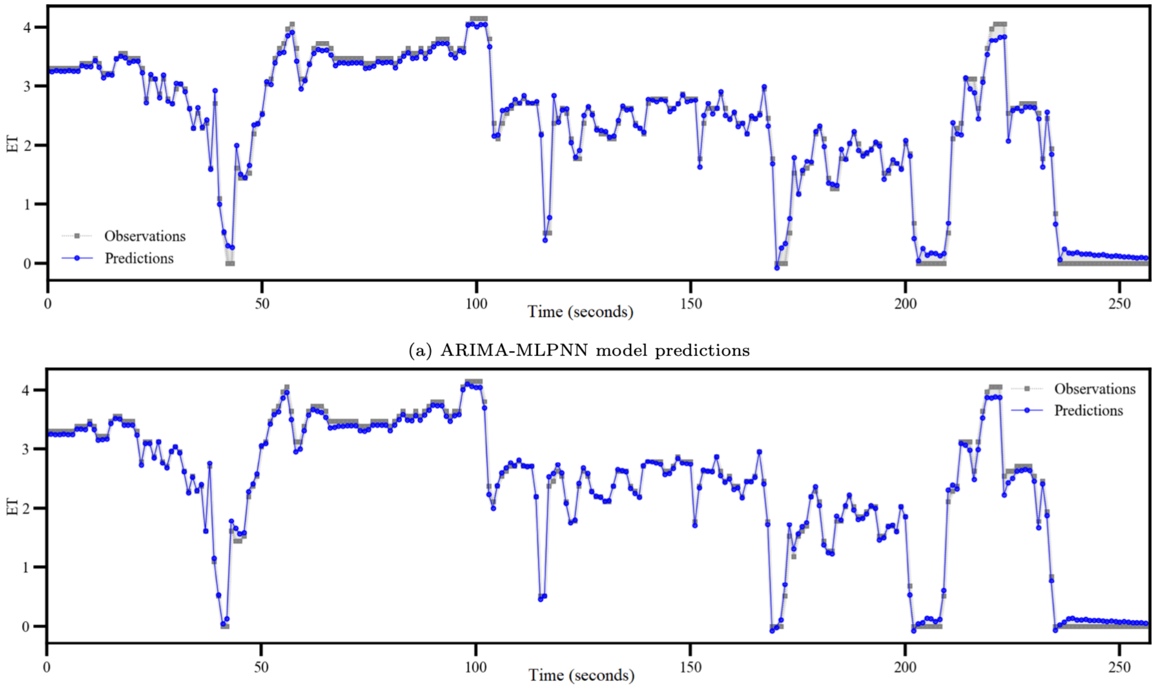
(a) ARIMA model predictions

(b) MLPNN model predictions

(c) WNN model predictions

Figure 3.5

Individual models predictions for ET subsystem

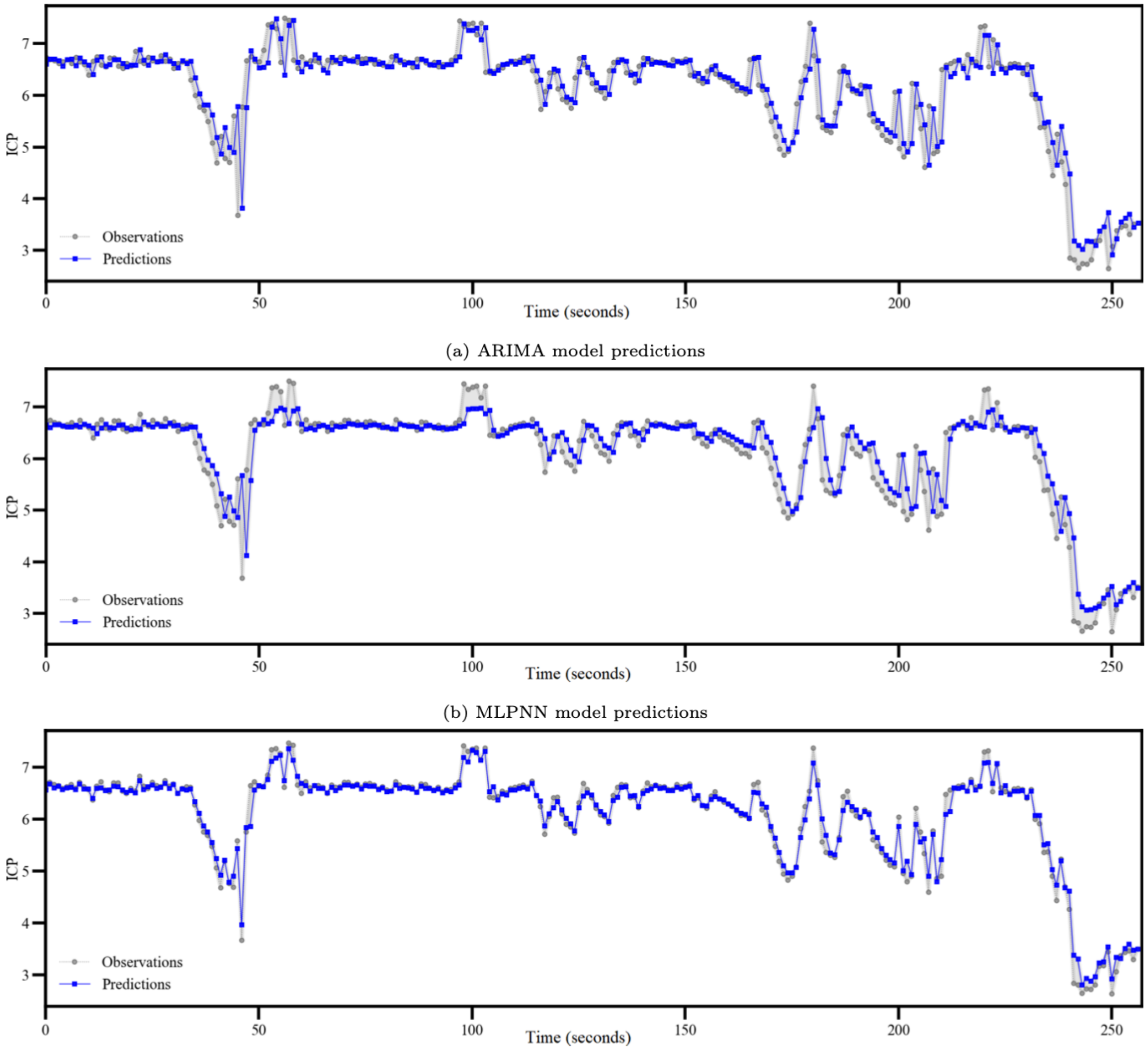


(a) ARIMA-MLPNN model predictions

(b) ARIMA-WNN model predictions

Figure 3.6

Hybrid ARIMA-MLPNN and ARIMA-WNN predictions for ET subsystem



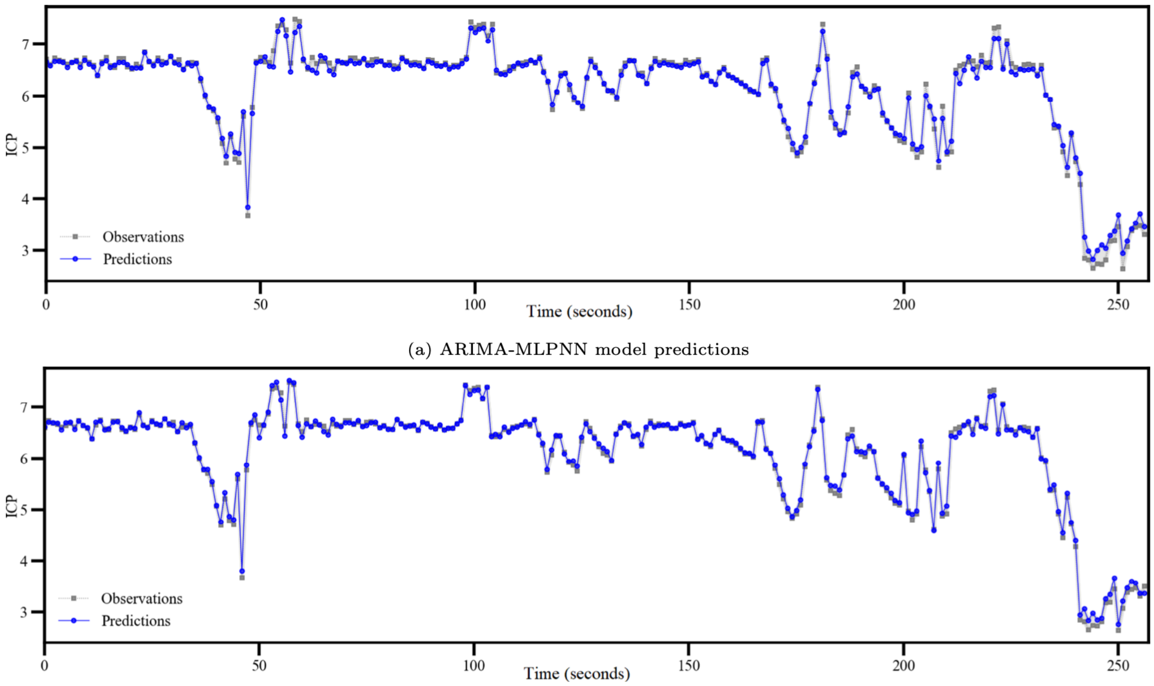
(a) ARIMA model predictions

(b) MLPNN model predictions

(c) WNN model predictions

Figure 3.7

Individual models predictions for ICP subsystem

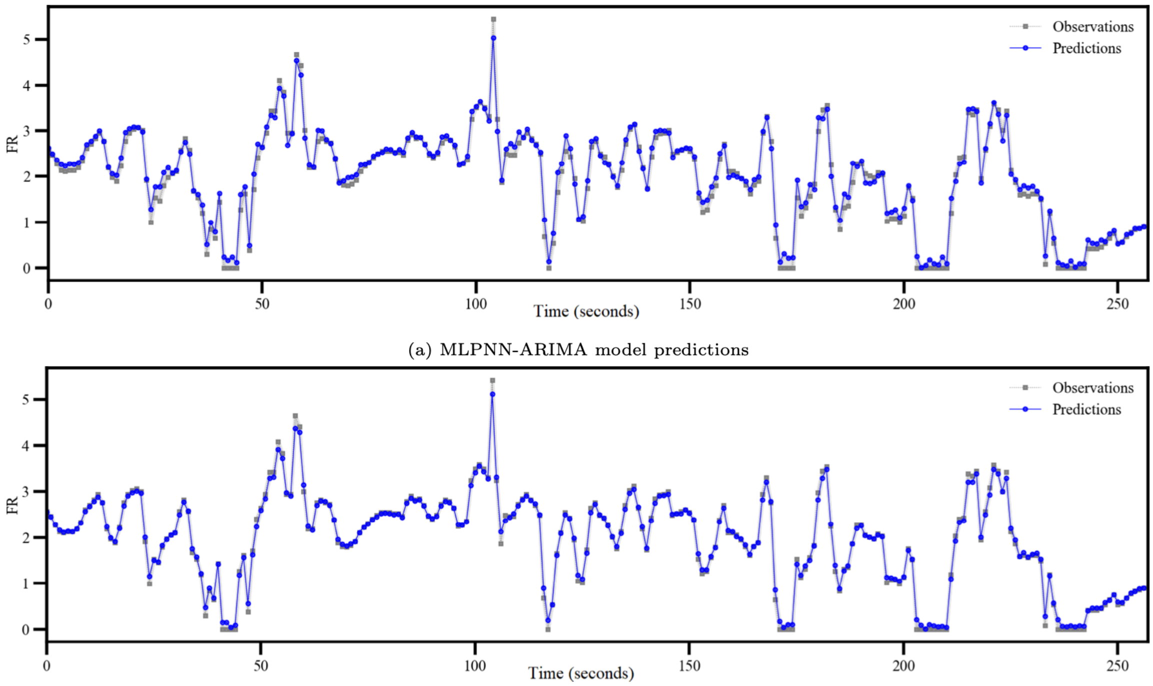


(a) ARIMA-MLPNN model predictions

(b) ARIMA-WNN model predictions

Figure 3.8

Hybrid ARIMA-MLPNN and ARIMA-WNN predictions for ICP subsystem

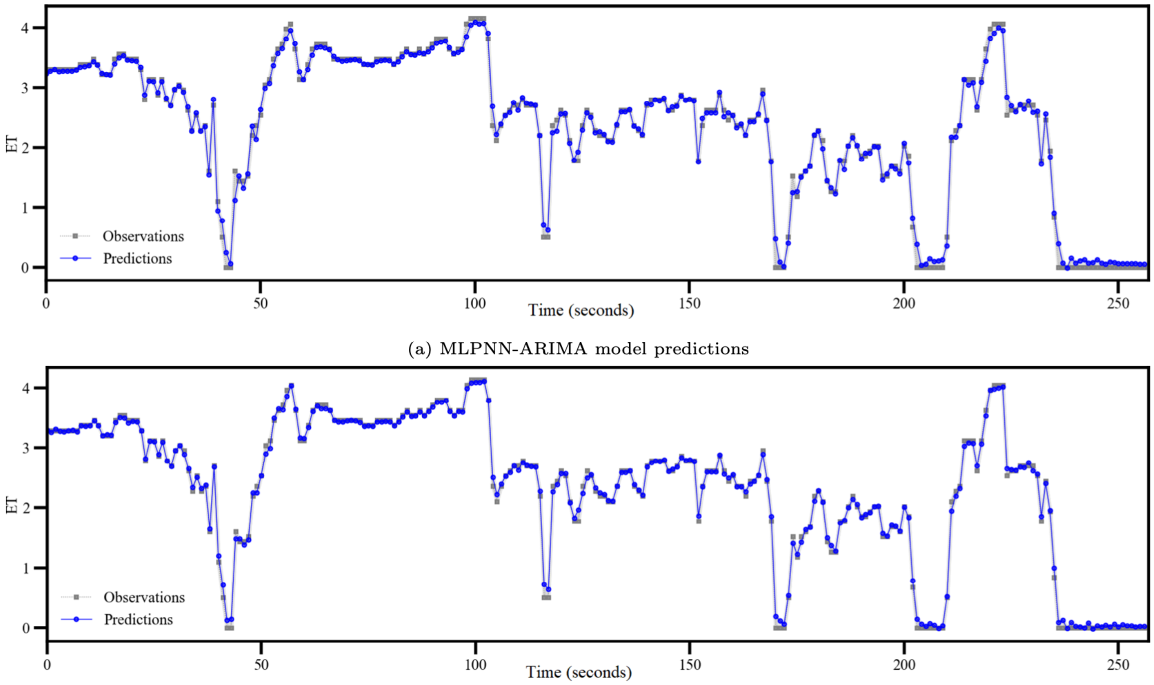


(a) MLPNN-ARIMA model predictions

(b) WNN-ARIMA model predictions

Figure 3.9

Hybrid MLPNN-ARIMA and WNN-ARIMA predictions for FR subsystem

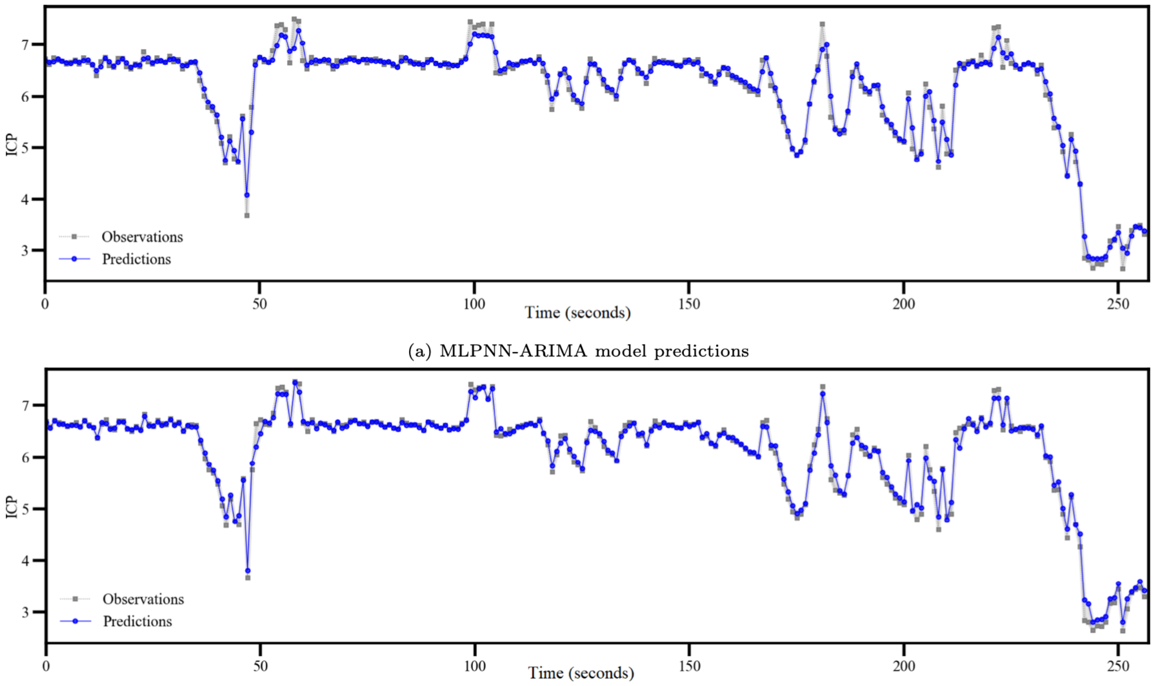


(a) MLPNN-ARIMA model predictions

(b) WNN-ARIMA model predictions

Figure 3.10

Hybrid MLPNN-ARIMA and WNN-ARIMA predictions for ET subsystem



(a) MLPNN-ARIMA model predictions

(b) WNN-ARIMA model predictions

Figure 3.11

Hybrid MLPNN-ARIMA and WNN-ARIMA predictions for ICP subsystem

Table 3.4

Forecasting efficiency of different models for FR, ET, and ICP subsystems

Sub-system		ARIMA		MLPNN		WNN		ARIMA-MLPNN		ARIMA-WNN		MLPNN-ARIMA		WNN-ARIMA	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
FR	MAE	39.15	36.37	28.57	26.66	19.87	20.79	15.00	15.16	18.86	15.20	11.91	11.28	8.90	5.44
	RMSE	0.59	0.53	0.38	0.35	0.27	0.27	0.2	0.2	0.23	0.18	0.16	0.15	0.13	0.08
	NSE	0.65	0.73	0.86	0.89	0.93	0.93	0.96	0.96	0.95	0.97	0.97	0.98	0.98	0.99
ET	MAE	19.91	24.68	9.87	14.34	7.75	9.05	8.55	8.76	9.81	6.31	5.72	6.54	3.95	4.08
	RMSE	0.39	0.4	0.15	0.2	0.1	0.12	0.11	0.12	0.14	0.08	0.08	0.1	0.06	0.06
	NSE	0.84	0.89	0.98	0.97	0.99	0.99	0.99	0.99	0.98	1.00	0.99	0.99	0.99	1.0
ICP	MAE	21.18	24.56	10.56	11.92	10.16	9.26	7.01	6.99	5.69	4.7	8.21	8.96	7.43	6.59
	RMSE	0.37	0.4	0.16	0.17	0.21	0.14	0.1	0.1	0.08	0.07	0.13	0.14	0.14	0.1
	NSE	0.86	0.84	0.97	0.97	0.96	0.98	0.99	0.99	0.99	1.00	0.98	0.98	0.98	0.99
Avg	MAE	26.75	28.54	16.33	17.64	12.6	13.03	10.19	10.3	11.46	8.74	8.61	8.93	6.76	5.37
	RMSE	0.45	0.44	0.23	0.24	0.19	0.18	0.14	0.14	0.15	0.11	0.12	0.13	0.11	0.08
	NSE	0.79	0.82	0.94	0.94	0.96	0.97	0.98	0.98	0.97	0.99	0.98	0.99	0.99	0.99

Table 3.5

Improvement percentage of models for subsystem FR

Indicator		ARIMA		MLPNN		WNN		ARIMA-MLPNN		ARIMA-WNN		MLPNN-ARIMA		WNN-ARIMA	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
MAE	MLPNN	27.0%	26.7%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	49.2%	42.8%	30.4%	22.0%	0.0%	0.0%	–	–	–	–	–	–	–	–
	ARIMA-MLPNN	61.7%	58.3%	47.5%	43.2%	24.5%	27.1%	0.0%	0.0%	20.5%	0.3%	–	–	–	–
	ARIMA-WNN	51.8%	58.2%	34.0%	43.0%	5.1%	26.9%	-20.0%	-0.3%	0.0%	0.0%	–	–	–	–
	MLPNN-ARIMA	69.6%	69.0%	58.3%	57.7%	40.1%	45.7%	20.6%	25.6%	36.9%	25.8%	0.0%	0.0%	–	–
	WNN-ARIMA	77.3%	85.0%	68.8%	79.6%	55.2%	73.8%	40.7%	64.1%	52.8%	64.2%	25.2%	51.8%	0.0%	0.0%
RMSE	MLPNN	35.5%	34.6%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	54.4%	49.6%	29.2%	22.9%	0.0%	0.0%	–	–	–	–	–	–	–	–
	ARIMA-MLPNN	65.7%	62.7%	46.8%	42.9%	24.8%	25.9%	0.0%	0.0%	10.9%	-11.2%	–	–	–	–
	ARIMA-WNN	61.5%	66.4%	40.3%	48.6%	15.7%	33.3%	-10.9%	10.0%	0.0%	0.0%	–	–	–	–
	MLPNN-ARIMA	72.2%	72.9%	56.9%	58.5%	39.2%	46.1%	19.1%	27.3%	27.9%	19.2%	0.0%	0.0%	–	–
	WNN-ARIMA	78.6%	85.7%	66.9%	78.1%	53.2%	71.5%	37.7%	61.6%	44.5%	57.3%	23.0%	47.1%	0.0%	0.0%
NSE	MLPNN	31.2%	20.7%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	42.2%	26.9%	8.4%	5.1%	0.0%	0.0%	–	–	–	–	–	–		
	ARIMA-MLPNN	47.1%	31.0%	12.1%	8.5%	3.4%	3.2%	0.0%	0.0%	1.1%	-0.8%	–	–	–	–
	ARIMA-WNN	45.4%	32.0%	10.9%	9.4%	2.3%	4.0%	-1.1%	0.8%	0.0%	0.0%	–	–	–	–
	MLPNN-ARIMA	49.2%	33.4%	13.8%	10.5%	4.9%	5.1%	1.5%	1.9%	2.6%	1.0%	0.0%	0.0%	–	–
	WNN-ARIMA	50.9%	35.3%	15.1%	12.1%	6.1%	6.6%	2.6%	3.3%	3.8%	2.5%	1.1%	1.4%	0.0%	0.0%

Table 3.6

Improvement percentage of models for subsystem ET

Indicator		ARIMA		MLPNN		WNN		ARIMA-MLPNN		ARIMA-WNN		MLPNN-ARIMA		WNN-ARIMA	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
MAE	MLPNN	50.5%	41.9%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	61.1%	63.4%	21.4%	36.9%	0.0%	0.0%	9.4%	-3.2%	21.0%	-30.2%	–	–	–	–
	ARIMA-MLPNN	57.0%	64.5%	13.3%	38.9%	-9.4%	3.2%	0.0%	0.0%	12.9%	-27.9%	–	–	–	–
	ARIMA-WNN	50.7%	74.4%	0.5%	56.0%	-26.6%	30.2%	-12.9%	27.9%	0.0%	0.0%	-41.7%	3.5%	–	–
	MLPNN-ARIMA	71.3%	73.5%	42.0%	54.4%	26.2%	27.7%	33.1%	25.3%	41.7%	-3.5%	0.0%	0.0%	–	–
	WNN-ARIMA	80.2%	83.5%	60.0%	71.6%	49.0%	54.9%	53.8%	53.4%	59.7%	35.4%	30.9%	37.7%	0.0%	0.0%
RMSE	MLPNN	63.0%	50.2%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	74.4%	70.2%	30.8%	40.1%	0.0%	0.0%	9.2%	-1.9%	26.5%	-34.3%	–	–	–	–
	ARIMA-MLPNN	71.8%	70.8%	23.8%	41.3%	-9.2%	1.9%	0.0%	0.0%	19.1%	-33.0%	–	–	–	–
	ARIMA-WNN	65.1%	80.4%	5.8%	60.7%	-26.5%	34.3%	-19.1%	33.0%	0.0%	0.0%	-38.4%	20.1%	–	–
	MLPNN-ARIMA	78.5%	75.5%	42.0%	50.8%	16.2%	17.8%	23.9%	16.2%	38.4%	-20.1%	0.0%	0.0%	–	–
	WNN-ARIMA	86.0%	85.6%	62.2%	71.1%	45.4%	51.7%	50.4%	50.8%	59.8%	26.5%	34.8%	41.2%	0.0%	0.0%
NSE	MLPNN	15.8%	9.6%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	17.1%	11.6%	1.1%	1.8%	0.0%	0.0%	0.2%	0.0%	0.9%	-0.6%	–	–	–	–
	ARIMA-MLPNN	16.9%	11.6%	0.9%	1.8%	-0.2%	0.0%	0.0%	0.0%	0.7%	-0.6%	–	–	–	–
	ARIMA-WNN	16.1%	12.2%	0.2%	2.4%	-0.9%	0.6%	-0.7%	0.6%	0.0%	0.0%	-1.2%	0.2%	–	–
	MLPNN-ARIMA	17.5%	11.9%	1.4%	2.2%	0.3%	0.3%	0.5%	0.3%	1.2%	-0.2%	0.0%	0.0%	–	–
	WNN-ARIMA	18.0%	12.4%	1.9%	2.6%	0.7%	0.8%	0.9%	0.8%	1.6%	0.2%	0.4%	0.4%	0.0%	0.0%

Table 3.7

Improvement percentage of models for subsystem ICP

Indicator		ARIMA		MLPNN		WNN		ARIMA-MLPNN		ARIMA-WNN		MLPNN-ARIMA		WNN-ARIMA	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
MAE	MLPNN	50.5%	41.9%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	61.1%	63.4%	21.4%	36.9%	0.0%	0.0%	9.4%	-3.2%	21.0%	-30.2%	–	–	–	–
	ARIMA-MLPNN	57.0%	64.5%	13.3%	38.9%	-9.4%	3.2%	0.0%	0.0%	12.9%	-27.9%	–	–	–	–
	ARIMA-WNN	50.7%	74.4%	0.5%	56.0%	-26.6%	30.2%	-12.9%	27.9%	0.0%	0.0%	-41.7%	3.5%	–	–
	MLPNN-ARIMA	71.3%	73.5%	42.0%	54.4%	26.2%	27.7%	33.1%	25.3%	41.7%	-3.5%	0.0%	0.0%	–	–
	WNN-ARIMA	80.2%	83.5%	60.0%	71.6%	49.0%	54.9%	53.8%	53.4%	59.7%	35.4%	30.9%	37.7%	0.0%	0.0%
RMSE	MLPNN	63.0%	50.2%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	74.4%	70.2%	30.8%	40.1%	0.0%	0.0%	9.2%	-1.9%	26.5%	-34.3%	–	–	–	–
	ARIMA-MLPNN	71.8%	70.8%	23.8%	41.3%	-9.2%	1.9%	0.0%	0.0%	19.1%	-33.0%	–	–	–	–
	ARIMA-WNN	65.1%	80.4%	5.8%	60.7%	-26.5%	34.3%	-19.1%	33.0%	0.0%	0.0%	-38.4%	20.1%	–	–
	MLPNN-ARIMA	78.5%	75.5%	42.0%	50.8%	16.2%	17.8%	23.9%	16.2%	38.4%	-20.1%	0.0%	0.0%	–	–
	WNN-ARIMA	86.0%	85.6%	62.2%	71.1%	45.4%	51.7%	50.4%	50.8%	59.8%	26.5%	34.8%	41.2%	0.0%	0.0%
NSE	MLPNN	15.8%	9.6%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	17.1%	11.6%	1.1%	1.8%	0.0%	0.0%	0.2%	0.0%	0.9%	-0.6%	–	–	–	–
	ARIMA-MLPNN	16.9%	11.6%	0.9%	1.8%	-0.2%	0.0%	0.0%	0.0%	0.7%	-0.6%	–	–	–	–
	ARIMA-WNN	16.1%	12.2%	0.2%	2.4%	-0.9%	0.6%	-0.7%	0.6%	0.0%	0.0%	-1.2%	0.2%	–	–
	MLPNN-ARIMA	17.5%	11.9%	1.4%	2.2%	0.3%	0.3%	0.5%	0.3%	1.2%	-0.2%	0.0%	0.0%	–	–
	WNN-ARIMA	18.0%	12.4%	1.9%	2.6%	0.7%	0.8%	0.9%	0.8%	1.6%	0.2%	0.4%	0.4%	0.0%	0.0%

Table 3.8

Average improvement percentage of different models

Indicator		ARIMA		MLPNN		WNN		ARIMA-MLPNN		ARIMA-WNN		MLPNN-ARIMA		WNN-ARIMA	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
MAE	MLPNN	38.9%	38.2%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	52.9%	54.3%	22.9%	26.1%	0.0%	0.0%	–	–	–	–	–	–	–	–
	ARIMA-MLPNN	61.9%	63.9%	37.6%	41.6%	19.1%	20.9%	0.0%	0.0%	15.4%	-9.1%	–	–	–	–
	ARIMA-WNN	55.0%	67.2%	26.3%	46.9%	4.4%	28.1%	-15.4%	9.1%	0.0%	0.0%	–	–	–	–
	MLPNN-ARIMA	67.8%	68.7%	47.3%	49.4%	31.6%	31.5%	15.5%	13.3%	28.4%	4.7%	0.0%	0.0%	–	–
	WNN-ARIMA	76.9%	83.4%	62.2%	73.1%	50.9%	63.6%	39.3%	54.0%	48.7%	49.4%	28.2%	46.9%	0.0%	0.0%
RMSE	MLPNN	49.3%	46.0%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	57.3%	60.4%	15.8%	26.7%	0.0%	0.0%	–	–	–	–	–	–	–	–
	ARIMA-MLPNN	69.7%	68.9%	40.3%	42.5%	29.1%	21.5%	0.0%	0.0%	19.0%	-14.2%	–	–	–	–
	ARIMA-WNN	62.6%	73.3%	26.3%	50.6%	12.5%	32.7%	-19.0%	14.2%	0.0%	0.0%	-25.9%	6.3%	–	–
	MLPNN-ARIMA	72.3%	71.5%	45.4%	47.3%	35.2%	28.1%	8.6%	8.4%	25.9%	-6.3%	0.0%	0.0%	–	–
	WNN-ARIMA	80.9%	85.0%	62.4%	72.2%	55.3%	62.1%	37.0%	51.7%	48.9%	43.7%	31.1%	47.3%	0.0%	0.0%
NSE	MLPNN	19.0%	14.7%	0.0%	0.0%	–	–	–	–	–	–	–	–	–	–
	WNN	21.8%	17.8%	4.6%	4.0%	0.0%	0.0%	–	–	–	–	–	–	–	–
	ARIMA-MLPNN	24.5%	19.3%	4.6%	4.0%	2.2%	1.8%	0.0%	0.0%	1.0%	-0.5%	–	–	–	–
	ARIMA-WNN	23.3%	19.9%	3.6%	4.6%	1.2%	1.8%	-1.0%	0.5%	0.0%	0.0%	-1.4%	0.0%	–	–
	MLPNN-ARIMA	25.0%	19.9%	5.0%	4.5%	2.7%	1.8%	0.4%	0.5%	1.4%	0.0%	0.0%	0.0%	–	–
	WNN-ARIMA	26.1%	21.2%	5.9%	5.7%	3.5%	2.9%	1.3%	1.6%	2.3%	1.1%	0.8%	1.1%	0.0%	0.0%

Table 3.9

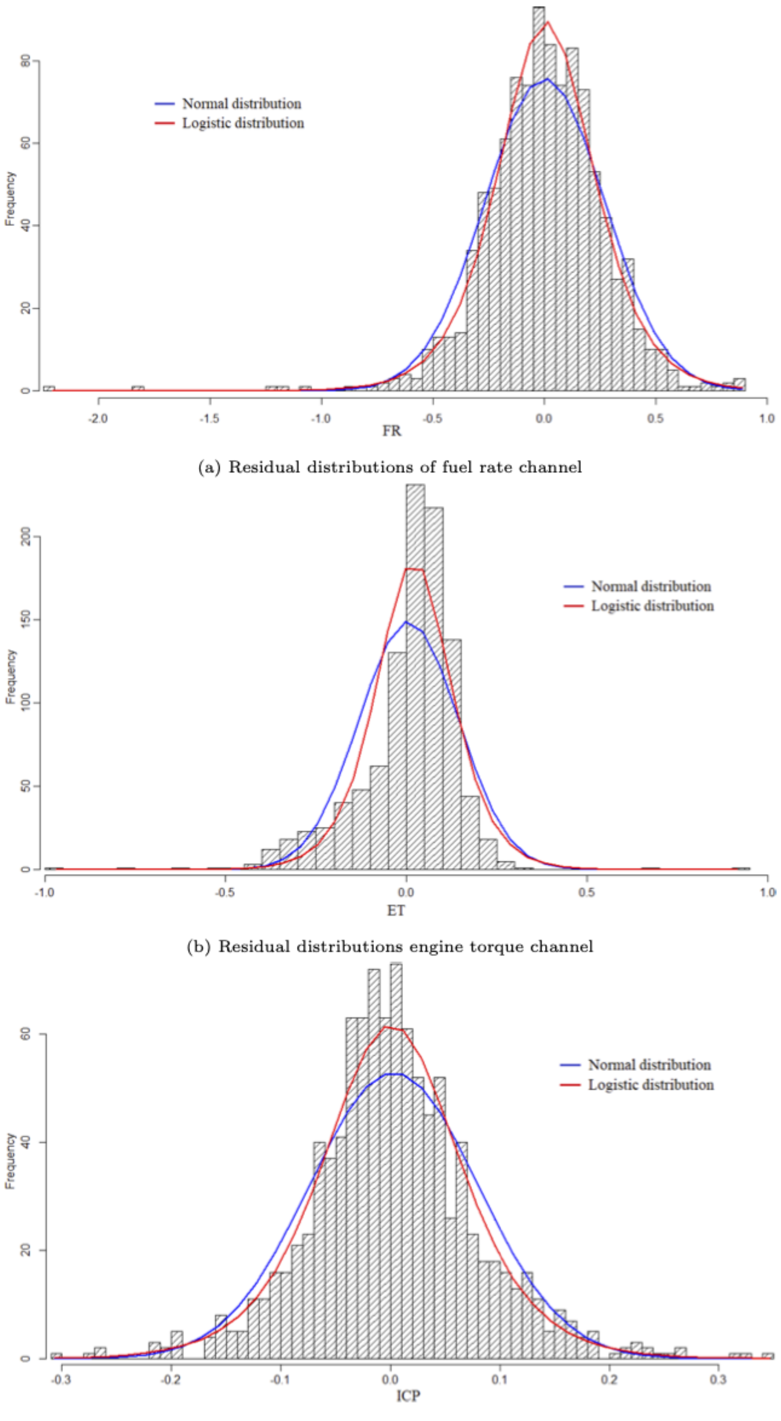
Distribution parameters and thresholds for residual of different subsystems

Sub-system	Normal distribution							Logistic distribution						
	μ	σ	LLV	AIC	T^c			μ	s	LLV	AIC	T^c		
					90%	95%	99%					90%	95%	99%
FR	-0.0012	0.126	667.78	-1331.55	0.277	0.371	0.579	0.0029	0.065	729.39	-1454.78	0.143	0.192	0.299
ET	-0.0002	0.055	1514.51	-3025.02	0.121	0.162	0.253	0.0009	0.029	1563.35	-3122.69	0.063	0.085	0.133
ICP	0.0011	0.142	547.24	-1090.48	0.311	0.417	0.651	0.0051	0.056	809.59	-1615.17	0.123	0.165	0.258

Table 3.10

False positive and false negative rates of Normal and Logistic distributions

Sub-system	Normal distribution						Logistic distribution					
	$CL = 90\%$		$CL = 95\%$		$CL = 99\%$		$CL = 90\%$		$CL = 95\%$		$CL = 99\%$	
	FP[1]	FN[2]	FP	FN	FP	FN	FP	FN	FP	FN	FP	FN
FR	0.0	0.186	0.0	0.194	0.0	0.194	0.023	0.137	0.0	0.160	0.0	0.194
ET	0.027	0.027	0.004	0.05	0.0	0.061	0.122	0.0	0.076	0.004	0.015	0.038
ICP	0.158	0.035	0.062	0.042	0.027	0.054	0.324	0.023	0.259	0.023	0.124	0.035
Avg	0.062	0.083	0.022	0.095	0.009	0.103	0.156	0.053	0.112	0.062	0.046	0.089



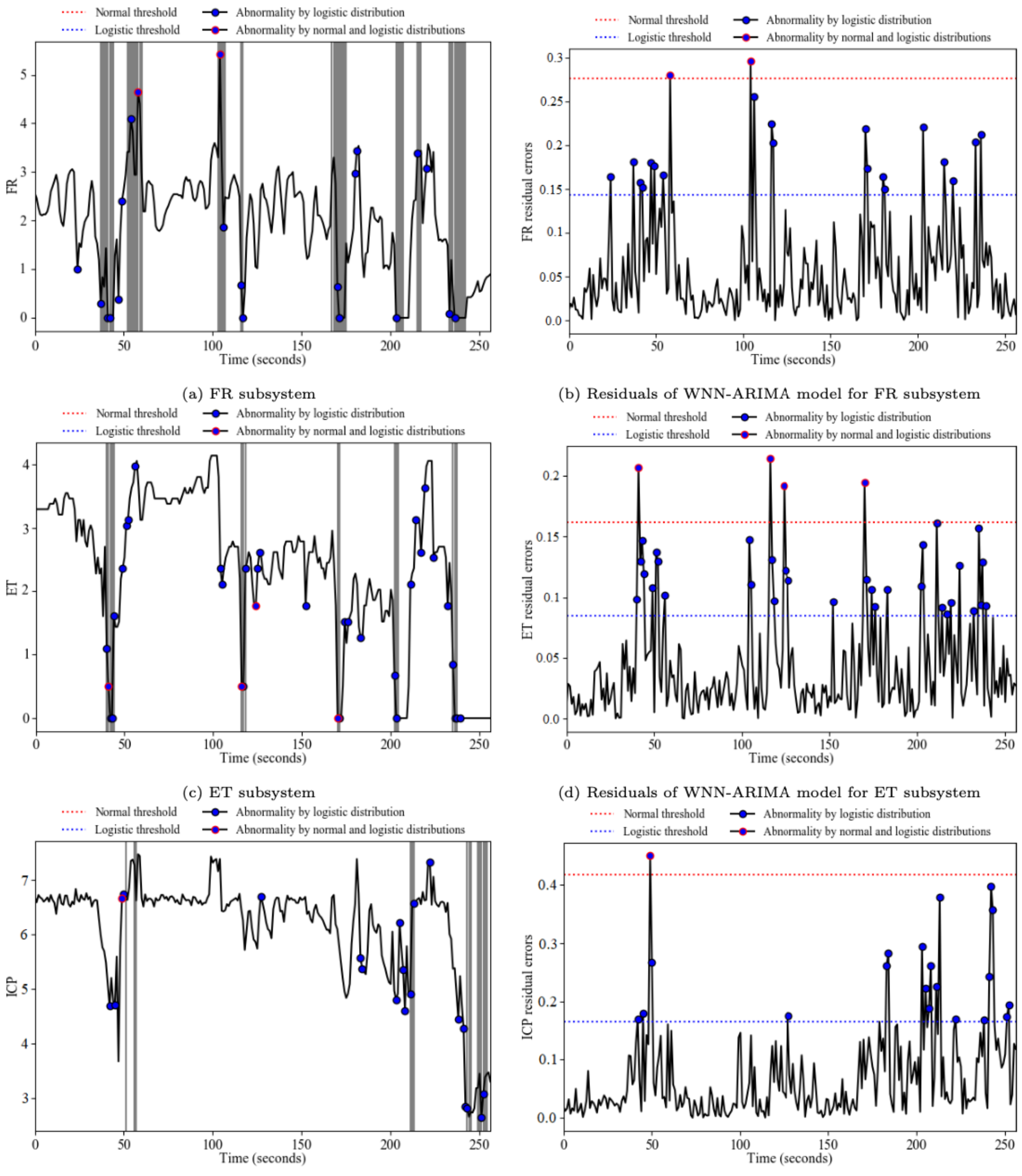
(a) Residual distributions of fuel rate channel

(b) Residual distributions engine torque channel

(c) Residual distributions of injection control pressure

Figure 3.12

Residual distributions for FR, ET, and ICP subsystems



(a) FR subsystem

(b) Residuals of WNN-ARIMA model for FR subsystem

(c) ET subsystem

(d) Residuals of WNN-ARIMA model for ET subsystem

(e) ICP subsystem

(f) Residuals of WNN-ARIMA model for ICP subsystem

Figure 3.13

Abnormal behavior detection for FR, ET, and ICP subsystems

CHAPTER IV

LEVERAGING MULTI-LAYER LONG SHORT TERM MEMORY AUTOENCODER AND ONE-CLASS SUPPORT VECTOR MACHINE FOR VEHICLE BEHAVIOR MODELING AND UNHEALTHY STATE DETECTION

4.1 Introduction

Nowadays, vehicles install a variety of complex subsystems that work together to achieve the desired performance. The failure of these complicated subsystems may result in expensive maintenance, a short life cycle, or even passenger casualties. Failure in a vehicle subsystem can be defined as an unexpected event that occurs at a certain timestamp and may lead to an afterward worse event or a series of other unexpected events. Although, failure in the performance of the vehicle subsystems can be identified early as an abnormal behavior before being detected as a critical issue, because vehicle performance will begin to degrade once a subsystem failure manifests. In other words, failures in a vehicle's components not only represent the abnormal signals but also emphasize the necessity of deploying maintenance procedures. Indeed, early detection of abnormal behaviors ensures a sufficient time interval for implementing appropriate countermeasures to prevent extra losses and sustain the nominal performance of the vehicle. Therefore, the timely monitoring of the performance of different subsystems is worthy of investigation as it helps to identify the imminent anomalous behaviors, extend vehicle life cycle, and therefore ensure passengers' safety. In this regard, the need for computer-aided diagnosis is vital to ensure accurate fault detection and system diagnosis in the complex subsystems of the vehicle [123].

Automated fault detection methods usually rely on the training and analysis of data that are obtained by various sensors attached to different subsystems of the vehicle. There are numerous types of sensors (e.g., engine speed sensor, oil temperature sensor, manifold absolute pressure sensor, etc.) installed into modern vehicles that are arranged to continuously send essential signals to monitor each subsystem. These sensors collect a massive amount of time series data that can be used to model and analyze the operational behavior of different functions of the vehicle. In this research, a multivariate model and anomaly detection method are developed to detect abnormalities in the operational behavior of different subsystems and identify the unhealthy states of the vehicle. This model monitors the health status of different subsystems and quickly recognizes the timely abnormalities or unexpected patterns based on multiple time series data.

Anomaly detection in multivariate time series sequences is one of the challenging problems in the technology era. Several studies in the literature worked on developing effective solutions for different real applications such as manufacturing, automotive, financial, etc.

4.2 Related Work

Anomaly detection problems can be categorized in different manners depending on the wide variety of scenarios and algorithms. One of the most common categorizations is based on the level of supervision in algorithms including supervised, semi-supervised, and unsupervised algorithms. Another categorization also exists based on machine learning and deep learning techniques. [32] conducted a comprehensive review of different anomaly detection algorithms and their limitations.

In this section, an overview of commonly used anomaly detection techniques for detecting point anomalies will be provided first. Then, anomaly detection algorithms developed for time series data will be presented. Finally, anomaly detection algorithms constructed based on deep neural networks will be discussed.

The *k*-Nearest Neighbors (*k*-NN) anomaly detection technique is one of the simplest and traditional methods for point anomaly detection [158]. This method is computationally intensive, highly dependent on *k* value, and may fail if a normal data point has not sufficient neighbors. Using *k*-NN, [28] introduced Local Outlier Factor (LOF) method for local density-based anomaly detection. This method assumes that the neighbors of data points are spherically distributed. In some applications, data points can be distributed in a linear manner. [179] proposed Connectivity based Outlier Factor (COF) to address the linear connection shortcoming of the LOF technique. The drawback of the COF algorithm is incorrect outlier detection when clusters with different densities are very close to each other which was resolved in Influenced Outlierness(INFLO) algorithm [85]. Clustering-based algorithms such as Cluster-Based Local Outlier Factor (CBLOF) are also used for unsupervised outlier detection.

Many semi-supervised and unsupervised types of anomaly detection algorithms are based on One-Class Support Vector Machine (OCSVM). [10] introduced the unsupervised version of OCSVM for anomaly detection purposes. This algorithm learns a decision boundary that maximizes the separation between the origin and data points. Anomaly data points fall outside the trained regions. Other developments of such algorithms include studies of [205] and [33]. [79] proposed a time series anomaly detection model based on the OCSVM algorithm. This technique defines six meta-features based on univariate and multivariate time series data and deploys the

OCSVM model on meta-features space for detecting abnormalities. [114] studied the Support Vector Data Description (SVDD) approach for detecting anomalies in uncertain data.

To characterize time series features, various anomaly detection techniques have been developed to identify anomalous patterns. Statistical AutoRegressive Moving Average (ARMA) model and its variations such as ARIMA and VARMAX have been widely leveraged for time series prediction and anomaly detection [209, 11, 8]. Recently, Long Short Term Memory (LSTM) networks have emerged as a powerful technique in time series analysis [110, 227, 44]. LSTM networks belong to the recurrent neural network category that exhibits great performance in handling sequential data [37]. Different LSTM architectures have been developed to analyze long-term dependencies and detect abnormalities in multiple applications such as turbofan engines [57, 50], hydraulic machinery [123], and machine life estimation [204]. [122] proposed stacked LSTM to detect time series anomalies. In this approach, an error threshold is defined based on the training errors on normal timestamps. Then, using the given threshold, a set of time series data is marked as normal or anomalous. [35] proposed a deep Recurrent Neural Network (RNN) model with LSTM units to predict Electrocardiography (ECG) signals and detect arrhythmia in the human heart. [89] combined wavelet and Hilbert transform with deep neural network to detect anomalies in earthquake activity. [97] presented a web traffic anomaly detection model by integrating Convolutional Neural Network (CNN), LSTM, and deep neural network. [202] integrated LSTM and Gaussian Bayes model for outlier detection in the Internet of Things (IoT). [189] developed a multi-layer Bi-Directional LSTM (BD-LSTM) model to identify anomalous events in complex surveillance scenes of smart cities. [17] proposed an LSTM model for performance anomaly detection on temporal irregularities in logs. [112] proposed an LSTM model to cooperatively predict outputs and identify anomalies in manufacturing processes. [39] studied an anomaly detection model for recognizing abnormalities in satellite telemetry data based on Bayesian deep learning without domain knowledge. [31] presented an aircraft track anomaly detection method by integrating Multidimensional Outlier Descriptor (MOD) and BD-LSTM.

Autoencoders are another type of unsupervised learning technique that leverages neural network framework to learn the latent patterns of input data. These models are trained to reproduce the input data as the expected output in which the reproduction errors are used as metrics for detecting abnormalities. Typically, autoencoder is used for dimension reduction or visualization purposes. Due to its considerable efficacy in data encoding, it has gained much attraction for anomaly and novelty detection activities [226, 9, 42]. Integrating autoencoders with LSTM networks as encoding and decoding units constructs a powerful architecture for learning sequential data and anomaly detection [122]. An LSTM encoding unit maps the input sequence to a latent vector representation and then the LSTM decoding unit reproduces the input sequence. [36] proposed BD-LSTM autoencoder for sequential anomaly detection in cybersecurity data. [231] studied a variational LSTM autoencoder model for anomaly detection in imbalanced industrial big data. Leveraging classical central limit theorem, [120] presented an enhanced LSTM autoencoder model for anomaly detection in unlabelled time series data. [139] proposed an LSTM autoencoder model for forecasting and anomaly detection in supply chain sale data.

4.3 Proposed ML-LSTMAE framework

In this paper, a Multi-Layer LSTM AutoEncoder (ML-LSTMAE) framework is proposed to monitor the operational behavior of different subsystems of a vehicle and identify unhealthy states by analyzing multivariate time series data. The vehicle system involves multiple functional subsystems such as engine, transmission, and fuel subsystems that operate individually or with other subsystems to satisfy customized requirements. Developing a deep neural network architecture, the proposed multivariate model learns the normal pattern of each subsystem and attempts to detect anomalous behaviors timely. The *temporal continuity* assumption plays a significant role in modeling the behavior of complicated system [2]. Based on this assumption, time series sequence will not be expected to change abruptly unless there are abnormalities. Thus, the idea is to identify abrupt changes that exhibit a lack of continuity. Furthermore, to represent the learned latent patterns in abstract and identify the abnormal moments of the operating vehicle, a heuristic algorithm is developed based on Principal Component Analysis (PCA) and OCSVM techniques.

4.3.1 Long Short-Term Memory Network

LSTM is a powerful recurrent neural network for modeling sequential data [75]. Retaining long-term dependencies in the data sequence, the LSTM network addresses the popular *vanishing gradient* problem in vanilla RNNs through multiplicative gated units. Different variations in LSTM networks have been studied in multiple applications such as time series analysis [91], speech recognition [213], and natural language processing [137]. An LSTM network consists of a chain of repeated modules of neural networks, each of them has three control gates: forget gate, input gate, and output gate. Each gate includes a sigmoid neural network layer and a pointwise multiplication operation. The sigmoid layer generates a number in the range of $[0, 1]$, indicating the portion of input information that is passed through. Let $\chi = \{x_t^c,\ \forall\, t \in \mathcal{T},\ c \in C\}$ denotes the sequence of input vectors and $x_t \in \mathbb{R}^C$ represents a C-dimensional vector of reading for C data channels of vehicle at timestamp t. Note that each data channel $c \in C$ involves the timely reading of a subsystem of the vehicle. The LSTM network can work with time series sequence of any of these data channels and its performance can vary depending on the input data. Given the new input sequence x_t in state t, the LSTM modules operate as following steps:

1. Forget gate: This gate decides what portion of information should be thrown away or kept. The information consisting of the previous hidden state information and current input data should be let through a sigmoid function that outputs a number in the interval $[0, 1]$ as follows:

$$f_t = \sigma(W_{xf}^T x_t + W_{hf}^T h_{t-1} + b_f) \quad \forall\, t \in \mathcal{T} \tag{4.1}$$

   where σ is the sigmoid activation function, h_{t-1} denotes previous output state, and W_f^T and b_f represent the weight matrix and bias term of the forget gate.

2. Input gate: Before storing the processed data, the input gate operates to update the cell state. It passes the previous hidden state and current input into a sigmoid function that determines the values to be updated by transforming the values to be between 0 and 1. Passing the previous hidden state and current input vector into the activation function $\mathcal{F}_c$ at the same time, the vector of candidate values $\tilde{C}_t$ is created. This vector helps to regulate the network by squishing the values between 1 and -1.

$$i_t = \sigma(W_{xi}^T x_t + W_{hi}^T h_{t-1} + b_i) \quad \forall\, t \in \mathcal{T} \tag{4.2}$$

$$\tilde{C}_t = \mathcal{F}_c(W_{xc}^T x_t + W_{hc}^T h_{t-1} + b_c) \quad \forall\, t \in \mathcal{T} \tag{4.3}$$

where W_{xi}^T and W_{hi}^T stand for the weight matrices and b_i shows the bias term for input gate vector i_t. Also, W_{xc}^T and W_{hc}^T represent the weights and b_c indicates the bias term for memory cell state $\tilde{C}_t$. Then, using pairwise multiplication $\odot$, the cell state vector is updated as follows:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad \forall\, t \in \mathcal{T} \tag{4.4}$$

3. Output gate: Finally, this gate determines the next hidden state value as follows:

$$o_t = \sigma(W_{xo}^T x_t + W_{ho}^T h_{t-1} + b_o) \quad \forall\, t \in \mathcal{T} \tag{4.5}$$

$$h_t = o_t \odot \mathcal{F}_h(C_t) \quad \forall\, t \in \mathcal{T} \tag{4.6}$$

where W_{xo}^T and W_{xo}^T represent the weight matrices and b_o shows the bias term for the output gate vector o_t. This gate determines the portion of the cell state that should be outputted. Also, $\mathcal{F}_h$ indicates the activation fuction for computing the hidden state h_t. The Rectified Linear Unit (ReLU) activation function is considered for both $\mathcal{F}_c$ and $\mathcal{F}_h$ operations. Moreover, the training phase of each LSTM unit is reguarlized by using an L2 (i.e., ridge) regularizer with penalty of 0.001.

Figure 4.1 on the following page represents the general architecture of LSTM network. In this network, the cell state runs straight down the entire chain and retains the sequential information which helps LSTM to gain knowledge from sequential timestamps. Various types of LSTM networks have been studied in the literature. [66] investigated popular types of LSTM networks and showed that these models are almost the same and only a few models perform more efficiently than others in some specific problems.

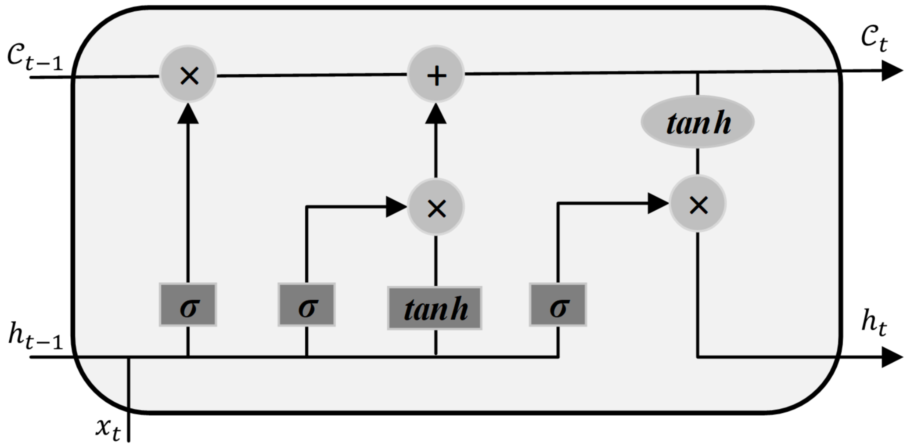


Figure 4.1

General architecture of LSTM

4.3.2 Autoncoder LSTM

Autoencoder is an unsupervised neural network that aims to minimize the reconstruction errors between input and output based on observational sequences of data. Learning a proper encoding-decoding scheme from training data leads to negligible reconstruction errors for predicting test data with statistically similar characteristics. This network holds advantages in detecting abnormality when new data with statistically different characteristics are fed into the model. The autoencoder network fails to appropriately reconstruct the discrepant data which results in a larger reconstruction error. This large error is the residual that indicates the abnormality. Indeed, the encoder-decoder scheme learns patterns from normal data sequences and can be used to identify abnormalities in multivariate time series. That is, it is trained for constructing only normal instances and when an anomalous sequence is fed, it may not be reconstructed well and results in a higher residual error. This high residual value has a practical meaning since abnormal data are not always available or it is impossible to reconstruct all of their variants.

In an autoencoder scheme, the input layer feeds data into the network, the encoder compresses data into latent space, and the decoder decompresses the encoded representation of data to the output. Comparing the encoded-decoded output to the input data, the residual error is propagated through the network architecture to update the network weights. Different variants of autoencoder networks have been studied in the literature such as vanilla autoencoder, regularized autoencoder, convolutional autoencoder, and LSTM autoencoder. The LSTM autoencoder is a recurrent model whose both encoder and decoder are LSTM networks. LSTM can learn patterns in long sequences of data that makes it become an adequate tool for multivariate time series forecasting and anomaly detection. Figure 4.2 on the next page represents the general architecture of the proposed multi-layer LSTM autoencoder network. The proposed network utilizes two layers of LSTM networks for encoding the data sequences and two layers of LSTM networks for decoding the encoded representation of data. Due to the complexity of learning the patterns in multivariate time series, the proposed architecture helps to develop an appropriate model to make accurate predictions and abnormal behavior detection in the data sequence of various subsystems of the vehicle.

4.3.3 One-Class Support Vector Machine

OCSVM was initially introduced by [166] for anomaly detection in high dimensional real-world data. This algorithm attempts to find the minimal subsets in the input space that comprises a predefined fraction of data. If the input data is mostly normal with a high predefined fraction, the algorithm reaches a support boundary around dense areas that represent the normal data [76]. Subsequently, any test data places on or inside the estimated boundary are labeled as a normal point; otherwise, it is labeled as anomalous.

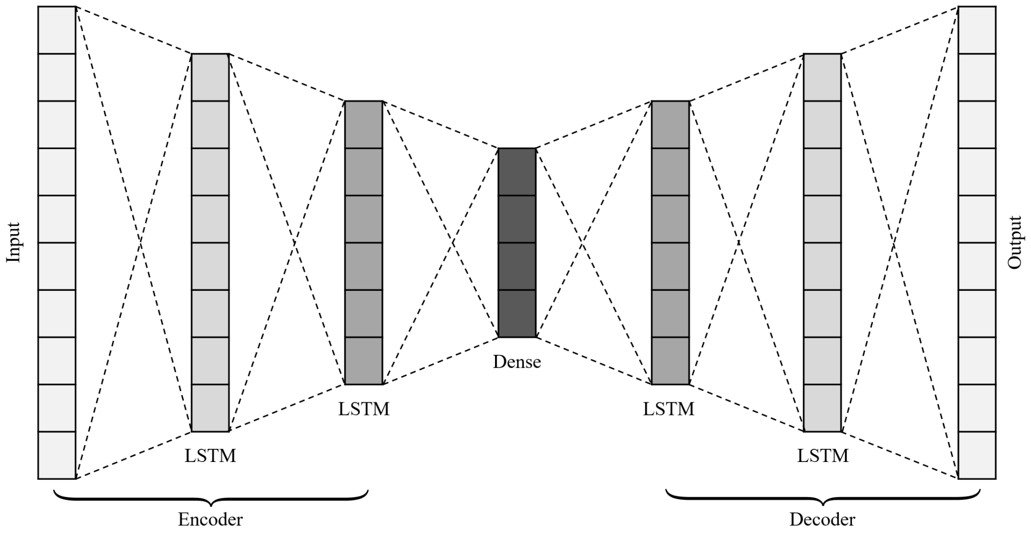


Figure 4.2

Architecture of proposed ML-LSTMAE

In many cases such as the real-world case study of this research, the distribution of training data is complicated and not linearly separable from the low-density areas in the input space. To address this challenge, OCSVM projects the training data $\{y_i, i \in N\}$, where $y_i \in \mathbb{R}^C$, from the input space $\mathbb{R}^C$ into a higher dimensional space $\mathbb{R}^{C'}$, where $C \ll C'$, by deploying a mapping function. The idea of feature mapping is to make the data linearly separable. A diverse range of mapping functions from a simple inner product to more sophisticated kernels can be used for various applications. With this technique, OCSVM determines a hyperplane that separates the projected data from the origin with the maximum possible margin. Thus, the problem of determining a nonlinear support boundary around the dense area in input space $\mathbb{R}^C$ is shrinking to find a linear boundary in the feature space $\mathbb{R}^{C'}$. The primal quadratic problem for finding the boundary can be defined as follows:

$$\min_{\omega,\xi,\rho} \frac{1}{2}\|\omega\|^2 + \frac{1}{\nu|N|}\sum_{i\in N}\xi_i - \rho \tag{4.7}$$

$$s.t. \quad \langle \omega \cdot \varphi(y_i) \rangle \geq \rho - \xi_i \quad \forall\, i \in N \tag{4.8}$$

$$\xi_i \geq 0 \quad \forall\, i \in N \tag{4.9}$$

where $\omega \in \mathbb{R}^C$ is a vector perpendicular to the separating hyperplane, $\xi = [\xi_1, \dots, \xi_N]$ stands for the vector of slack variables, ρ is a bias term that controls the distance to the origin, and $0 < \nu \leq 1$ is an upper bound on the fraction of margin errors and a lower bound on the fraction of support vectors. The slack variables allow some input data to fall on the origin side of the support boundary. This relaxation results in a solution with a greater margin from the origin and shrinks the estimated boundary around the normal data in the input space. The ν parameter adjusts the level of relaxation. If ν is set closed to 0, it forces $\xi_i \to 0$, $\forall i \in N$ and so vanishes the penalty term $\sum_{i\in N}\xi_i$ from the objective function 4.7. In other words, this forces the algorithm to learn a hyperplane that separates almost all training data from the origin in the feature space $\mathbb{R}^{C'}$. Conversely, setting the ν parameters closed to 1 allows more freedom to the algorithm to leave more points in the origin side of the hyperplane in which almost all training data are classified as anomalies in the final solution.

Since the explicit computation of the mapping function φ is extremely intensive, a similarity matrix is usually used in the OCSVM algorithm to project the data from input space $\mathbb{R}^C$ to feature space $\mathbb{R}^{C'}$. Instead, a kernel function $k(y_i, y_j)$ can be used as an efficient alternative to compute a positive-definite matrix $\mathcal{K} \in \mathbb{R}^{|N|\times|N|}$. In this dissertation, the well-known Gaussian kernel (also called Radial Basis Function(RBF)) is used as follows:

$$k(y_i, y_j) = exp(-\frac{\|y_i - y_j\|^2}{2\sigma^2}) \tag{4.10}$$

where $\sigma > 0$ shows the bandwidth parameter of the kernel. The distance between two projected instances y_i and y_j in the feature space can be defined as follows:

$$\begin{aligned} \|\varphi(y_i) - \varphi(y_j)\|^2 = k(y_i, y_i) + k(y_j, y_j) - 2k(y_i, y_j) = \\ 2\left[1 - exp(-\frac{\|y_i - y_j\|^2}{2\sigma^2})\right] \end{aligned} \tag{4.11}$$

Equation 4.7 shows a positive proportional relationship between $\|\varphi(y_i) - \varphi(y_j)\|$ and $\|y_i - y_j\|$. This implies that the Gaussian kernel preserves the ranking order of the distances between data points in the input and feature spaces. Using Lagrangian multipliers and the kernel function, [166] showed that the primal problem 4.7-4.9 can be converted to the following dual problem:

$$\min_{\alpha_i} \frac{1}{2} \sum_{i,j \in N} \alpha_i \alpha_j k(y_i, y_j) \tag{4.12}$$

$$s.t. \sum_{i \in N} \alpha_i = 1 \quad \forall\, i \in N \tag{4.13}$$

$$0 \leq \alpha_i \leq \frac{1}{\nu N} \quad \forall\, i \in N \tag{4.14}$$

Then, using the dual formulation 4.12-4.14, the value of ρ is computed by choosing a random data point y_i residing on the hyperplane as $\rho = \sum_j \alpha_j k(y_j, y_i)$. After the training phase, any unseen data point y can be predicted using the following formulations:

$$f(y) = sign(\langle \omega \cdot \varphi(y) - \rho \rangle) \tag{4.15}$$

$$f(y) = sign(\sum_j \alpha_j k(y_j, y) - \rho) \tag{4.16}$$

where equations 4.15 and 4.16 formulate the primal and dual based predictions, respectively. Consequently, a given solution divides the training data into three groups as follows:

1. Support Vectors (SVs): This group shows data points residing on the support boundary with $f(y) = 0$.

2. Nonsupport Vectors (NSVs): This group represents data point fall in inside the normal boundary with $f(y) > 0$.

3. Anomalies: This group shows data points locating outside the boundary with $f(y) < 0$.

Then, the proposed approach for multivariate time series forecasting and anomaly deletion are explained in the next section.

4.3.4 Anomaly Detection with ML-LSTMAE

Multivariate time series comprises a set of sequential data with more than one time-dependent variable. That means each variable depends not only on its past values but also on the values of other variables. These dependencies in multivariate time series can be used for modeling the interdependencies and forecasting the future values. Developing an accurate multivariate model for time series forecasting is a difficult task in many applications such as sensors data in the operating vehicle. [195] conducted a comprehensive study on advanced forecasting models for multivariate time series based on statistical approaches. Recently, with rapid developments of artificial neural networks, various LSTM networks have been proposed for resolving either difficult or out-of-scope problems to handle with conventional time series predictive models [174].

This study develops an LSTM-based method for monitoring the behavior of different subsystems (e.g., engine, transmission, fuel systems, etc.) of an operating vehicle and alerting unhealthy states. The proposed model involves modeling multivariate time series in an autoencoder architecture with multi encoding and decoding layers, recognizing abnormal behaviors of different subsystems, and determining the health status of the vehicle. As mentioned earlier, let $x_t = \{x_t^1, x_t^2, \ldots, x_t^C\}, t = 1, \ldots, \mathcal{T}$ represents the vehicle's multi-channel data at time t. In the vehicle's sensors data, each data channel $c \in C$ indicates the time series data collected by a specific sensor such as engine oil pressure, fuel rate, transmission oil temperature, etc. The proposed multi-layer LSTM network is trained based on the sequence of observations $\{\mathbf{x}_1, \ldots, \mathbf{x}_{|N|}\}$ where N denotes the set of observed data. Initially, individual observations are scaled using a Min-Max scalar as follows:

$$x^c_{t,scaled} = \frac{x^c_t - x^c_{min}}{x^c_{max} - x^c_{min}} \quad \forall\, c \in C,\ t \in \mathcal{T} \tag{4.17}$$

where x^c_t and $x^c_{t,scaled}$ stand for the original and the scaled values of data channel $c \in C$ at timestamp $t \in \mathcal{T}$, respectively. For the sake of simplicity, the scaled observations are denoted by x^c_t. Moreover, x^c_{min} and x^c_{max} denote the maximum and minimum values of data channel $c \in C$. A sliding window of size m, $m < N$ is used in the training process. That means m consecutive multivariate data are fed into the LSTMAE model simultaneously. Then, using the sliding window, the proposed LSTMAE model can read and encode the input sequence $\mathbf{X}_i = \{\mathbf{x}_t, \ldots, \mathbf{x}_{t-m+1}\}$ and regenerate the output $\widehat{\mathbf{X}}_i = \{\widehat{\mathbf{x}}_t, \ldots, \widehat{\mathbf{x}}_{t-m+1}\}$, $i = m+1, \ldots, |N|$. This process continues until the sliding window reaches the end of the training data. In this dissertation, each encoding step and decoding step of the algorithm consists of two stacked layers of LSTM networks. Figure 4.3 on page 145 illustrates the operations of the proposed ML-LSTMAE model with a sliding window of size $m = 2$. Let

λ, τ, and β denote learning rate, number of epochs, and training batch size. Moreover, γ_1 and γ_2 represent the number of LSTM units in the first and second encoding layers, and γ_3 and γ_4 show the number of LSTM units in the decoding layers. In real applications like operating vehicle data, flagging the unhealthy states correctly is more important than alerting the healthy states incorrectly. Due to the complexity of modeling multiple time series of different data channels, the stacked layers in encoding and decoding steps help to achieve more accurate predictions and decrease the false-negative rate in detecting anomalous behaviors of vehicle subsystems. Using the Mean Absolute Error (MAE) loss function, the weights of ML-LSMTAE are trained as follows:

$$L = \sum_{i=m+1}^{|N|} e_i \tag{4.18}$$

where the prediction error vector is computed as $e_i = \widehat{\mathbf{X}}_i - \mathbf{X}_i$, $i = m + 1, \ldots, |N|$. The Adam optimization technique with a learning rate of 0.01 is used to train this network. Then, the trained network predicts the unseen data, and the prediction errors are further analyzed for abnormality detection. The proposed ML-LSTMAE algorithm is trained only on the normal sequences of data and then the learned model is used for detecting abnormalities of multivariate time series. In this scheme, the multi-layer encoding-decoding model has only seen the normal data during the training phase and learned to reconstruct them. Once an abnormal data sequence is fed into this model, it may not be able to reconstruct it well and results in higher prediction errors. This shows practical meanings since abnormal data occasionally occur and it is impossible to cover all types of them. Thus, by evaluating the prediction errors in the time series of various data channels, the abnormal behaviors of different subsystems are flagged. To achieve the best prediction performance, it is necessary to optimize the parameters of the proposed model including the learning rate, the number of cells, and the dropout parameters. In some applications, choosing an appropriate sliding window is also a challenge. The capability of LSTM networks in learning long temporal dependencies in data makes this algorithm not need to pre-determine the size of the sliding window. In other words, it can find the optimal look-back data sequence on its own. In this dissertation, a variant of different sizes is tested for determining the best sliding window in training the proposed model on multivariate data channels.

Then, the residual errors of predictions made by the ML-LSTMAE model for test data are further analyzed to detect anomalous states of the vehicle. Several studies in the state of the art assumed that the test errors follow a Gaussian distribution and then the maximum likelihood estimation technique was used to estimate the parameters of the distribution [121]. However, assuming the Gaussian distribution for prediction errors may not be true in some practical applications. To tackle this challenge, a common machine learning algorithm (i.e., OCSVM) is used in this research that does not require any specific assumption on data. OCSVM is one of the powerful machine learning algorithms developed for anomaly detection purposes. Since applying the proposed ML-LSTMAE algorithm eliminates the dependencies in the multivariate time series data of different channels, the error vectors e_i can be considered as independent values. Then, the OCSVM algorithm can learn a supporting hyperplane to distinguish the normal and abnormal instances of data. Pseudocode 4 shows the steps of the proposed ML-LSTMAE model for predicting multivariate time series and the OCSVM model for detecting abnormalities.

Algorithm 4: ML-LSTMAE model

```
$Input$ : $\{\mathbf{x}_1, \ldots, \mathbf{x}_{|N|}\}$, $\lambda$, $\tau$, $\beta$, $m$, $\gamma_l$
Initialize parameters of ML-LSTMAE
for $iteration \leftarrow 1$ to $\beta$ do
    $\widehat{\mathbf{X}}_i = \{\widehat{\mathbf{x}}_t, \ldots, \widehat{\mathbf{x}}_{t-m+1}\} \leftarrow$ output of
      ML-LSTMAE($\mathbf{X}_i = \{\mathbf{x}_t, \ldots, \mathbf{x}_{t-m+1}\}$), $i = m + 1, \ldots, |N|$
    Compute prediction error vector $e_i = \widehat{\mathbf{X}}_i - \mathbf{X}_i$, $i = m + 1, \ldots, |N|$
    Compute loss function $\mathcal{L} = ||\widehat{\mathbf{X}}_i - \mathbf{X}_i||$
    Optimize parameters of ML-LSTMAE (i.e., $\beta$, $\gamma_l$) based on loss value $\mathcal{L}$ by
      using backpropagation method with learning rate $\lambda$
end
Optimize parameters of OCSVM based on prediction error vector
  $e_i$, $i = m + 1, \ldots, |N|$
Predict test data using trained ML-LSTMAE
Classify test errors using trained OCSVM
$Output$ : The classified data as normal and anomalous
```

4.4 Operating Vehicle Multiple Channel Time Series Data Wrangling

4.4.1 Data Description and Challenges

This case study investigates the operational data of 101 time series data channels that were recorded in seconds by different sensors of a specific vehicle between 2013/01/01 and 2014/03/31. Each data channel represents the performance records of a particular subsystem of the vehicles such as engine oil pressure, fuel rate, and transmission oil temperature. The challenges of working with this data include high dimensionality, combined operation status, and missing values. To evaluate the efficiency of the proposed ML-LSTMAE model, this data set is divided into a training set including multivariate time series of different channels between 2013/01/01 and 2013/06/30 and a test set comprises multivariate data channels from 2013/07/01 through 2014/03/31. We also know that the vehicle was working normally throughout the training set and it started abnormal behavior in the range of the testing data set a comprehensive maintenance effort has been made on the vehicle during this range. Thus, this research aims to develop a predictive model on the normal training data and detect abnormalities in the test data. This approach helps decision makers to predict the abnormal behavior of the vehicle and apply appropriate maintenance efforts at the right time that prevents extra losses and extends the vehicle life cycle. The multivariate time series data of this study are wrangled based on the following steps:

1) All data channels $c \in C$ with at least 20% missing values throughout all timestamps are excluded from the analysis. Thus, 21 data channels such as unsprung mass, roll angels, and relative speed of front axles are removed from the main data set.

2) All data channels $c \in C$ with a constant value or very low standard deviation are excluded from the analysis. Thus, 53 channels such as vehicle brake dynamic control, clutch switch, and brake switch are removed.

3) Since the multivariate data are continuously collected, missing values of different data channels can be recorded. If a missing value is at the beginning or the end of the cycle, it is replaced by the nearest non-null value; Otherwise, it is imputed using the linear interpolation method.

4) If there are two duplicate data channels, a channel with a higher percentage of missing values is removed.

4.4.2 Channel Data Preparation

Leveraging the preprocessing phase, the missing values of different data channels are addressed. It also reveals that not all of the data channels are helpful to model the behavior of the vehicle. Hence, the main data set is modified to include the most useful data channels. This step excludes data channels that are not only helpful for developing a more accurate model but also increase the computational complexity of implementing the proposed model. Consequently, 21 time series data channels are selected as the most effective data channels (e.g., transmission output shaft speed, engine oil pressure, engine coolant temperature, etc.) to implement the proposed ML-LSTMAE model for predicting the behavior of the vehicle and detecting the unhealthy states.

The proposed ML-LSTMAE model is programmed in python 3.7.3 environment on TensorFlow 2.4.1 as backend and Keras as core model development library. The OCSVM algorithm is also programmed in the python environment to detect abnormalities in the behavior of the vehicle. Besides, a desktop with 3.6GHz Intel(R) Core(TM) i7 7700 and 64GB of RAM is used as our computation platform.

4.5 Experimental Results

In this section, the performance of the proposed ML-LSTMAE network combining with the OCSVM algorithm is evaluated based on a real case study data and the NASA bearing data set. The NASA data set is available in NASA Prognostics Data Repository[1]. Initially, the proposed methods are used to predict the patterns and detect abnormalities in the NASA bearing data. Then, we use

[1] NASA Bearing Data Set, https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/bearing

the proposed approaches to forecast the behavior of the operating vehicle and detect unhealthy states based on a multivariate series of real data channels.

4.5.1 NASA Multivariate Bearing Data

In this data set, the vibration sensor reading were taken on four bearings that ran into failure under constant load over multiple days. It consists of individual files that represent 1-second vibration signal snapshots recorded at 10 minutes intervals. Each file involves 20,480 sensor data points per bearing with a sampling rate of 20 kHz. It is assumed that the bearings are mechanically degraded gradually over time. Then, the reading of different bearings are aggregated by using the mean absolute value of each 10 minutes vibration records. The aggregated data set is divided into a training set and a test set. Figure 4.4 on page 145 represents the training data set that includes sensor reading with the normal condition while Figure 4.5 on page 146 shows the test set that contains the abnormal reading that lead to the bearing failure. Furthermore, 5% of the training set is considered as a validation set during the training phase. The objective is to identify the anomalous behaviors in the bearing test set by using data in the training set. In other words, the training data are fed to train the ML-LSTMAE network, then the trained model is used to detect the abnormalities in the test set. Table 4.1 on the next page represents the optimal parameters of the trained ML-LSTMAE network. In this training, the learning rate, number of epochs, and training batch size are considered as $\lambda = 0.01$, $\tau = 100$, and $\beta = 10$, respectively. Moreover,

Figure 4.6 on page 146 represents a comparison between the MEA loss function of the ML-LSTMAE model on the training and validation data sets.

Table 4.1

Optimal configuration of ML-LSTMAE for NASA bearing data set

Layer	Output Shape	Number of Parameters
Input	(None, 1, 4)	0
LSTM 1	(None, 1, 20)	2000
LSTM 2	(None, 5)	520
Repeat Vector	(None, 1, 5)	0
LSTM 3	(None, 1, 5)	220
LSTM 4	(None, 1, 20)	2080
Output	(None, 1, 4)	84

This plot indicates a significant decrease in both training and validation errors after 100 epochs. Table 4.2 on the next page shows the prognostic performance of the proposed model for all four bearings in the training set. This table also confirms the superior performance of the proposed ML-LSTMAE network in learning the latent normal behaviors in the four bearings of the NASA data set with low deviations between the bearing reading and predicted values.

The proposed model learns the normal behaviors of different bearings such that a significant deviation from this model is considered abnormal behavior. Then, utilizing the trained model, the behaviors of four bearings are predicted in the test data sequences. Given the prediction errors vector in test data $e_i = \widehat{\mathbf{X}}_i - \mathbf{X}_i$, $i = m + 1, \ldots, |N|$, the absolute value of residuals are used to construct the support boundary that determines the normal behavior or abnormal behavior of the system. Figure 4.7 on page 147 illustrates the distribution of the training residuals.

Using the absolute value of the training residuals generated by the proposed ML-LSTMAE, the OCSVM technique is then used to build the support boundary. Training the OCSVM model with an RBF kernel and a variant of parameters, the optimal hyper-parameters $\nu = 0.001$ and

Table 4.2

Prognostic performance of proposed model for training bearings

Bearing	MAE	SSE	RMSE
Bearing 1	0.115	8.799	0.14
Bearing 2	0.071	3.905	0.0937
Bearing 3	0.051	2.024	0.067
Bearing 4	0.0095	0.085	0.014
Average	0.0618	3.703	0.0789

$\gamma = 5$ are considered to construct the support boundary that correctly classifies more than 0.99% of the training instances. Using the training support boundary, the prediction error vectors e_i for the testing phase are classified to recognize the abnormal behaviors of the bearing system. Leveraging the PCA method, the classification output of the OCSVM algorithm are reduced into a two-dimensional coordinate plane in Figure 4.8 on page 148.

As shown in Figure 4.8 on page 148, the characteristics extracted from the abnormal data tend to be in a group different than the attributes extracted from the normal data. This fact confirms the efficiency of the proposed ML-LSTMAE network in properly learning the essential parameters and format rules of the input data to reconstruct it. Consequently, the OCSVM algorithm can correctly classify the normal and abnormal behaviors of the bearing system.

Furthermore, Figure 4.9 on page 148 and Figure 4.10 on page 149 represent the anomaly behaviors (i.e., red points) of bearing 1 and bearing 4 which are detected by the proposed approach in the test time series data. For instance, there are some unusual high values in bearing 1 reading between 2004/16/02 and 2004/19/02 and in bearing 4 reading between 2004/17/02 and 2004/18/02, and in 2004/19/02. As shown in these figures, the ML-LSTMAE predicted the values of these bearings which are highly deviated from their actual observations. Subsequently, the OCSVM algorithm flagged these timestamps as abnormal points.

Comparison metrics True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) are used to evaluate the accuracy of the proposed ML-LSTMAE for different bearings time series. Note that the TP and TN represent the number of anomalies and normal instances that are correctly diagnosed by the OCSVM algorithm. The FP stands for the number of instances that are not correctly detected, and FN shows the incorrectly determined normal events. Using these metrics, the Precision (PR) and Recall (RE) measure the accuracy and the completeness of the results, respectively. Finally, F-score checks the balance between Precision and Recall measures.

$$PR = \frac{TP}{TP + FP} \tag{4.19}$$

$$RE = \frac{TP}{TP + FN} \tag{4.20}$$

$$AC = \frac{TP + TN}{TP + FP + TN + FN} \tag{4.21}$$

$$F - score = 2 \times \frac{PR \times RE}{PR + RE} \tag{4.22}$$

Table 4.3 on the next page represents the obtained results for analyzing the performance of the proposed LSTM-based prediction model and OCSVM classification algorithm. As this table shows, the proposed LSTM autoencoder method has considerable performance in reconstructing the input data that leads to accurate abnormality detection in the behavior of the bearing system. Indeed, leveraging the OCSVM algorithm results in anomaly detection with 0.98% accuracy and an F-score of 0.98%. These results verify the efficiency of the proposed ML-LSTMAE network combining with the OCSVM algorithm to flag the abnormal behavior of different bearings sequences in the NASA bearing data set.

Table 4.3

Performance analysis of the proposed anomaly detection method

Data	TP	TN	FP	FN	PR	RE	AC	F-score
NASA Bearing	451	82	5	5	0.989	0.989	0.982	0.989

4.5.2 Operating Vehicle Multiple Channel Time Series Data

In this section, the proposed ML-LSTMAE integrating with OCSVM algorithms are used to learn the normal patterns in the multivariate time series of an operating vehicle and recognize the unhealthy states of the vehicles. The characteristics of this multivariate data set are described in section 4.4.1. These time series data recorded by the sensors on different subsystems of the vehicle are often complex, nonlinear, period-dependent, and inter-correlated. It is also difficult to visually measure the health status of the operating vehicle based on multiple time series data of different subsystems. Therefore, developing a multivariate time series model is necessary to analyze these complex data streams appropriately. For instance, Figure 4.11 on page 149 and Figure 4.12 on page 150 represent sample training streams of the Transmission Output Shaft Speed (TOSS) and Transmission Gear Value (TGV) subsystems, respectively. As it is shown, high variations in the time series data of these subsystems require a multivariate model that is able to learn these sensitive trends.

Dividing the data streams into training and test data sets, 5% of the training set is used to validate the performance of ML-LSTMAE during the training procedure. The optimal configuration of the ML-LSTMAE for training on the multivariate data channels of the vehicle is presented in

Table 4.4

Optimal configuration of ML-LSTMAE for operating vehicle

Layer	Output Shape	Number of Parameters
Input	(None, 1, 21)	0
LSTM 1	(None, 1, 96)	45312
LSTM 2	(None, 36)	19152
Repeat Vector	(None, 1, 36)	0
LSTM 3	(None, 1, 36)	10512
LSTM 4	(None, 1, 96)	51072
Output	(None, 1, 21)	2037

Table 4.4. Moreover, the learning rate, number of epochs, and batch size during the training phase are considered as $\lambda = 0.001$, $\tau = 300$, and $\beta = 100$, respectively.

Figure 4.13 on page 150 compares the decrements of the lost function of the proposed model on the training and validation data sets. As this figure shows, both training and validation errors are reasonably decreased after 300 epochs.

Table 4.5 on the following page represents the prognostic performance of the ML-LSTMAE for learning the normal behaviors in the data channels of different subsystems of the vehicle.

As this table shows, the proposed model can learn the latent pattern in the training data set with low error metrics. Then, the OCSVM algorithm is trained on the residual errors of predicting the training data with the proposed ML-LSTMAE network. This algorithm constructs a support boundary that can be used for distinguishing the healthy and unhealthy states of the operating vehicle in test data streams. Thus, the trained model is used to predict the behavior of the vehicle in the test data. That is, the prediction errors of the ML-LSTMAE model on the test data are further analyzed by the constructed boundary in the training phase to identify the unhealthy states of the

Table 4.5

Prognostic performance of ML-LSTMAE in learning normal data of different subsystems

Data channel	MAE	SSE	RMSE
Engine Torque	0.0027	12.84	0.0044
Engine Load	0.0012	5.65	0.0029
Engine Speed	0.0017	9.89	0.0038
TOSS	0.0021	13.12	0.0060
TGV	0.0074	10.83	0.0128
Injection Control Pressure	0.0022	14.72	0.0047
Engine Coolant Temperature	0.0027	13.73	0.0045
Booster Pressure	0.0009	11.80	0.0042

vehicle. The test data includes the time series records of various data channels before and after implementing the maintenance requirements. For instance, Figure 4.14 on page 151 and Figure 4.15 on page 151 respectively show the sample test data before implementing the maintenance efforts for TOSS and TGV data channels, which exhibit abnormal behaviors with different ranges of values compared with the same training samples. Furthermore, Figure 4.16 on page 152 and Figure 4.17 on page 152 illustrate the sample test streams after deploying the maintenance efforts for these data channels. These figures represent the normal behaviors of the vehicle's subsystems similar to the corresponding sample training data.

The OCSVM algorithm is trained on the absolute values of the training errors for 21 data channels. This technique constructs a support boundary by using an RBF kernel function and the optimal hyper-parameters $\nu = 0.002$ and $\gamma = 20$. Similar to the NASA bearing data case, the developing support boundary in the vehicle multivariate time series classifies more than 99% of the training instances correctly. Then, to identify the unhealthy states of the vehicle, the trained OCSVM algorithm is used to classify the test instances as healthy and unhealthy points. In other words, the absolute values of the prediction errors in the test data are analyzed to identify the abnormalities. Hence, any timestamp with a test error that falls inside the constructed boundary is labeled as a normal point in which the vehicle was operating under a healthy state. Conversely, any timestamp with test residual falls outside the support boundary is labeled as an abnormal timestamp where the vehicle operated under an unhealthy state.

To illustrate the performance of the proposed method for detecting the unhealthy states of the vehicle, the PCA technique extracted the principal components of multivariate time series errors in the test set. Figure 4.18 on page 153 represents the variance ratio explained by the principal components.

Since the first two principal components explain 90% of the variance in the test errors, we can choose these two components to express the multivariate test errors in terms of these two new variables. Leveraging this transformation, the classification outputs of the OCSVM algorithm can be presented by these two principal components. Figure 4.19 on page 154 shows the healthy/unhealthy classifications of the vehicle's behavior in a two-dimensional coordinate plane.

In this figure, the RBF kernel is shown by the red circle that surrounds the normal range of the residual errors. Indeed, this kernel is trained to cover more than 99% percent of the training errors. The gray points show the prediction errors obtained in the training phase and the dark-gray points with a black edge represent the normal test errors. These two sets of errors fall inside the support boundary and thus indicate the healthy state of the operating vehicle. The red points are the test errors that fall outside the boundary and reveals the unhealthy states of the vehicle. Clearly, the test errors with unhealthy states are categorized in a different group comparing to the training errors and test errors with healthy states. Similar to the NASA bearing data set, this fact verifies the efficiency of the proposed ML-LSTMAE network to learn the essential pattern in the input data. As a result, applying the subsequent OCSVM algorithm can differentiate the healthy and unhealthy states of the vehicle's operations adequately.

Figure 4.20 on page 154 and Figure 4.21 on page 155 show the comparison between the samples of real values and predictions of ML-LSTMAE for time series data channels of TOSS and TGV subsystems. Additionally, the unhealthy states of these data channels are flagged by the red points. Obviously, the proposed prediction model catches the normal changing trends in these data channels. But, there are some abnormal behaviors in these subsystems before implementing the maintenance efforts that lead to unhealthy states of the vehicle. That is due to the fact that the ML-LSTMAE network was trained on a normal multivariate data channels set. When new data with statistically different characteristics are fed into the model, the proposed model fails to reconstruct them properly. This results in a large prediction error that indicates the presence of abnormality..

Moreover, Figure 4.22 on page 155 and Figure 4.23 on page 156 represent the predictions and actual observations for the sample TOSS and TGV test data channels after implementing the required maintenance efforts. As these figures confirm, after deploying the necessary maintenance, these subsystems returned to normal behavior with statistical characteristics analogous to the training time series. Therefore, no unhealthy states are reported on these slots of the test TOSS and TGV data channels.

These results verify the accuracy of the proposed approach in modeling the behaviors of different subsystems of the operating vehicle. Consequently, the abnormal behaviors are early predicted as unhealthy states of the vehicle long time before detecting them as critical operational issues. It helps to be aware of imminent failures before arising and allows appropriate time intervals to implement the best countermeasures which avoid extra losses.

4.6 Summary and limitation

This study presents a Multi-Layer LSTM Autoencoder (ML-LSTMAE) model for predicting the behavior of the multivariate time series data recorded by multiple sensors on different subsystems of the operating vehicle. Leveraging the One-Class Support Vector Machine (OCSVM) algorithm and Principal Component Analysis (PCA), an anomaly detection technique is developed for determining the healthy and unhealthy states of the vehicle. The nonlinear complexity of the latent patterns in the time series data of various subsystems challenges the tasks of behavior prediction and unhealthy states detection based on multivariate records. Developing a deep neural network architecture, the proposed approach learns the normal patterns in data channels of different subsystems and timely recognizes the unhealthy states of the vehicle. This study develops an LSTM based network to model the long-term dependencies in time series sequences of different subsystems. Moreover, an autoencoder scheme is combined with the proposed multi-layer LSTM network to minimize the reconstruction error between the input and output data sequences. This structure becomes advantageous in detecting unhealthy states of the vehicle when a new record with statistically different characteristics is fed into the model. Finally, the absolute values of training errors are used to construct a support boundary by the OCSVM algorithm to differentiate the healthy and unhealthy states of the vehicle.

To validate the efficiency of the proposed prediction model and anomaly detection algorithm, we applied two data sets: operating vehicle data set and NASA bearing data set. The operating vehicle data set includes 15 months time series records with 101 data channels, and the NASA bearing data set includes a large number of 20,110,378 observations for 4 bearing time series sequences. Compared with operating vehicle data set, NASA bearing data set is cleaner and more organized, which will be more obvious to show the performance of the proposed approach. Both results verify the performance of the proposed prediction and anomaly detection methods. Training the proposed prediction method on the operating vehicle train data set leads to an MAE value of 0.0008. The trained support boundary on the training errors of this data set surrounds 0.99% of the training instances. Using the learned boundary the test data of the operating vehicle are labeled as healthy and unhealthy states adequately. The obtained results represent that the proposed approach was able to successfully alert the unhealthy states of the vehicle before implementing the required maintenance. It helps to be aware of the imminent failures before detecting them as critical issues. NASA bearing data set confirms the performance of the proposed approach. Results show that the proposed ML-LSTMAE model fitted well to the training data of the NASA bearing data set with an MAE value of 0.061. Besides, training the OCSVM algorithm on the training errors of this data set classifies the 0.99% of the training instances correctly. That is, the developed OCSVM technique learned the normal patterns in the training data appropriately. Applying the trained support boundary on the NASA test data results in 0.98% accuracy for detecting the anomalous behaviors of the bearing system.

The managerial insights of the proposed prediction and anomaly detection models are presented as follows. Firstly, the proposed ML-LSTMAE network is able to predict the behavior of each individual subsystem of the vehicle which results in achieving a better understating of the vehicle's operating status. Consequently, it enables decision makers to provide more accurate recommendations regarding either predictive or preventive maintenance plans of the vehicle. Moreover, the proposed OCSVM-based anomaly detection framework triggers timely alerts for warning the imminent failures to the system that helps to mitigate the vehicle's operating risk and avoid extra costs. Additionally, the proposed approach of integrating the ML-LSTMAE prediction network and OCSVM anomaly detection algorithm will provide a general multivariate time series analysis mechanism that can be applied to any field with abnormal forecasting and detection need.

Despite the novel contributions of this study to the state of the art, there are some other research opportunities that deserve further analysis. First of all, the LSTM unit of the proposed prediction model can be further improved by deploying convolutional operations that accelerate the learning process by rapid extraction of meaningful information from the input data. Next, transfer learning can be leveraged to address the time-varying challenges latent in the time series sequences of different subsystems. Finally, evaluating the performance of other modeling approaches such as Bidirectional Encoder Representations from Transformers (BERT) for predicting the behavior of the operating vehicle and detecting anomalies is worthy of investigation. Future studies will consider these limitations and broaden the scope of this research.

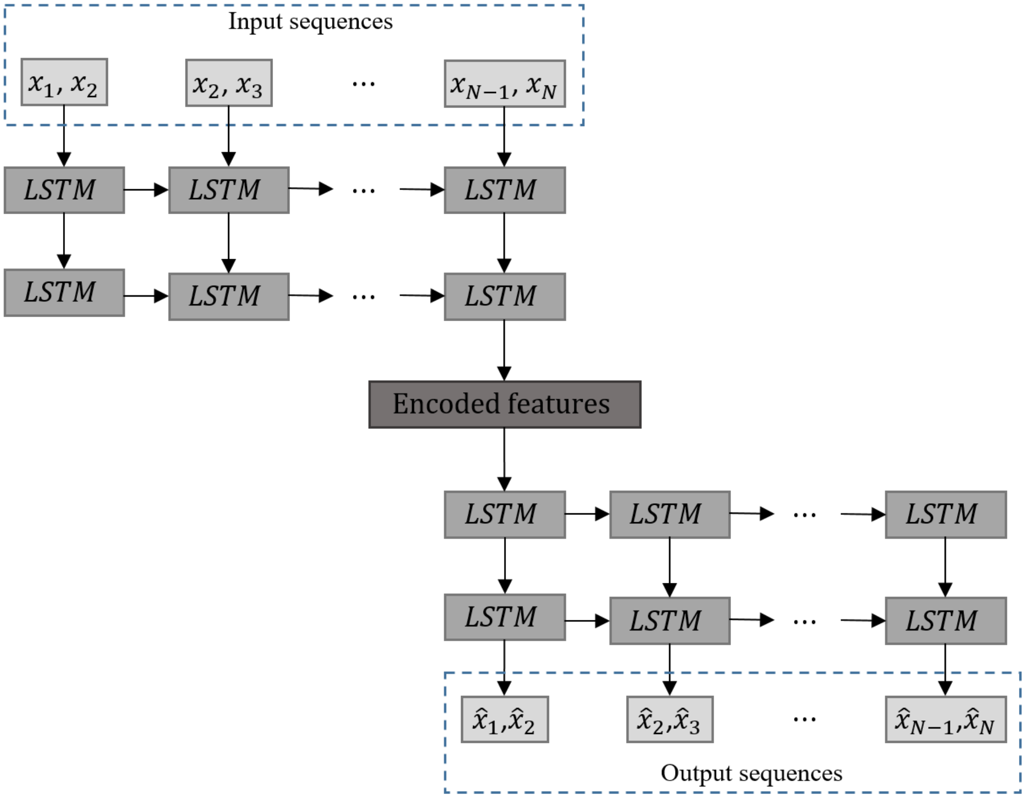


Figure 4.3

Operational scheme of the proposed ML-LSTMAE with sliding window of size 2

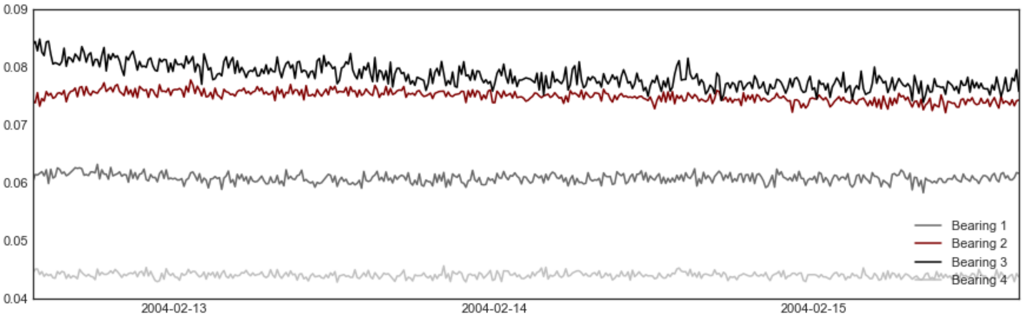


Figure 4.4

NASA bearings multivariate time series training data

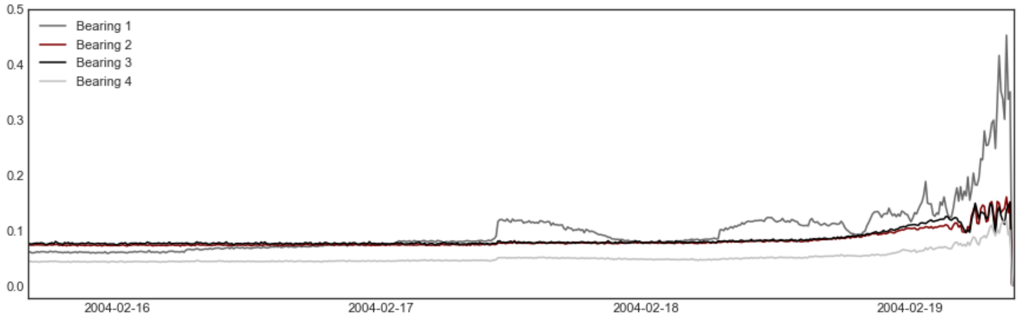


Figure 4.5

NASA bearings multivariate time series test data

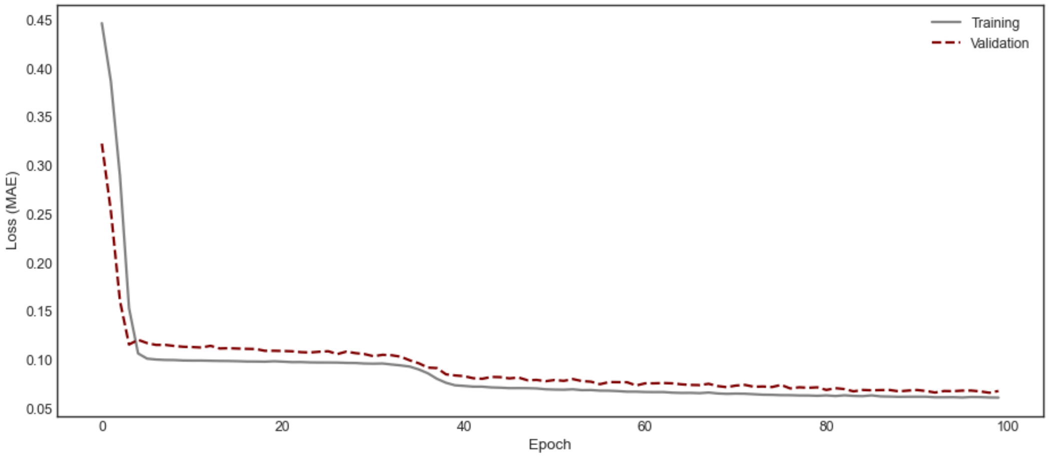


Figure 4.6

NASA bearing data set: training and validation loss plots

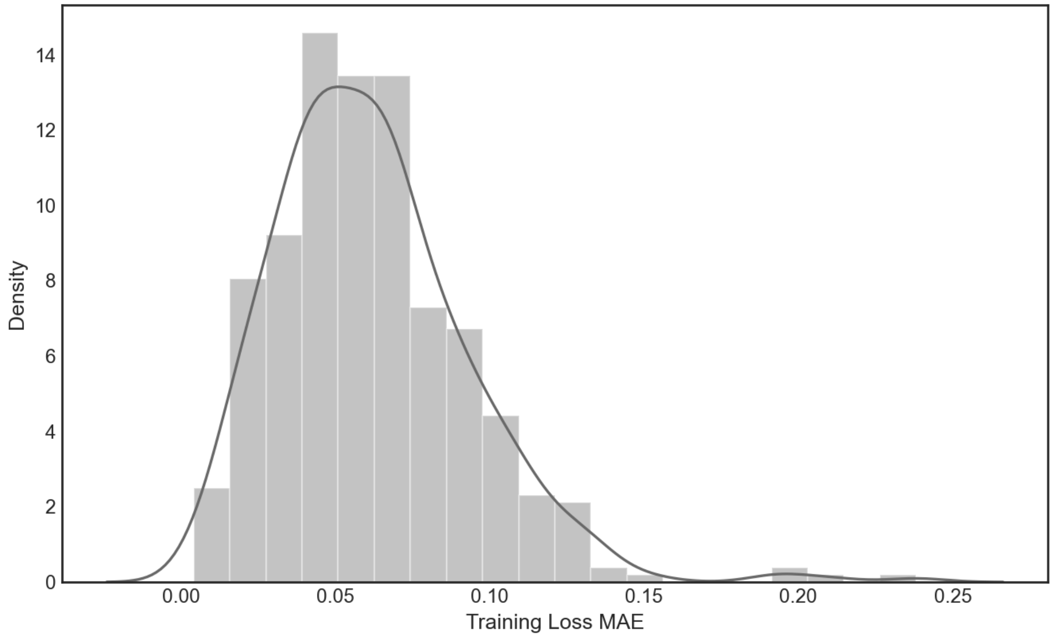


Figure 4.7

Distribution of errors in training data set

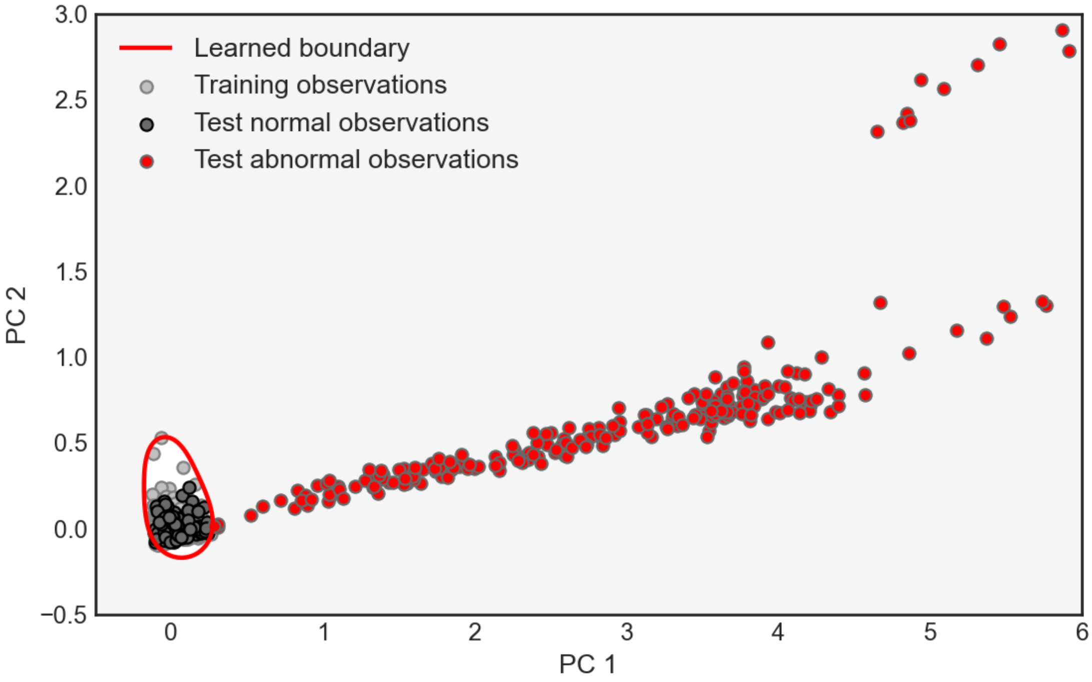


Figure 4.8

Normal/abnormal illustration of bearings test data

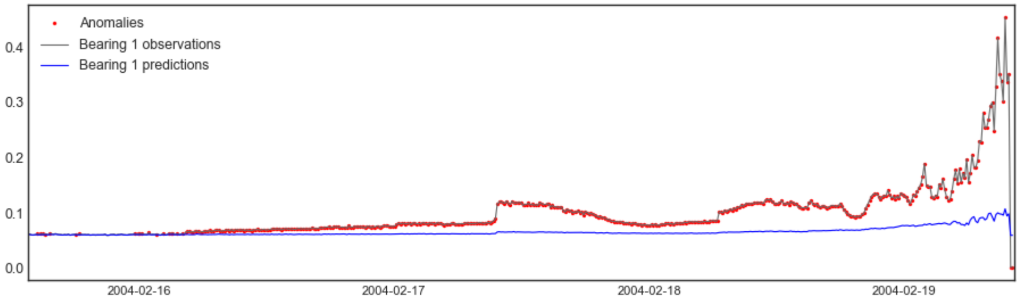


Figure 4.9

NASA Bearing 1: test data, predictions, and anomalies

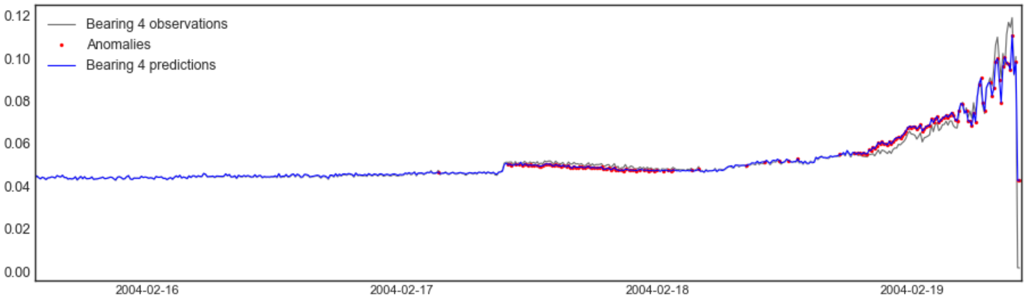


Figure 4.10

NASA Bearing 4: test data, predictions, and anomalies

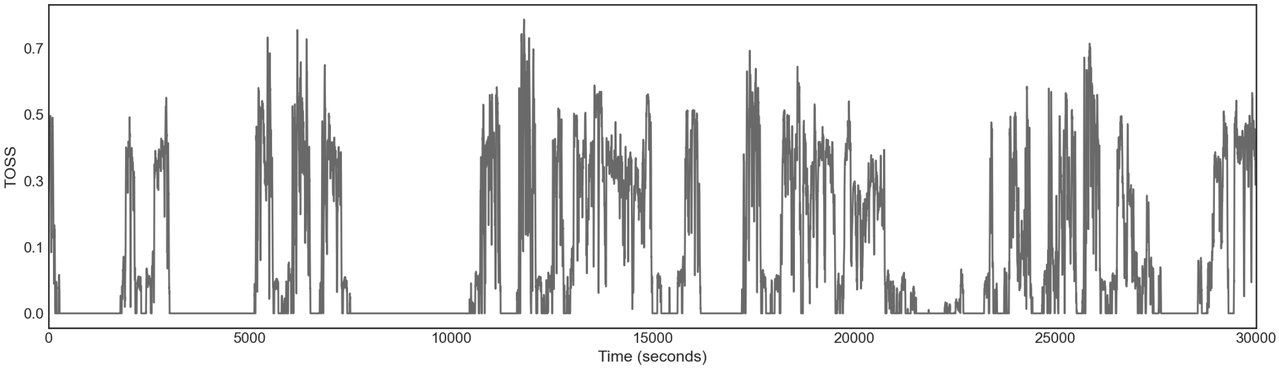


Figure 4.11

Sample TOSS training data

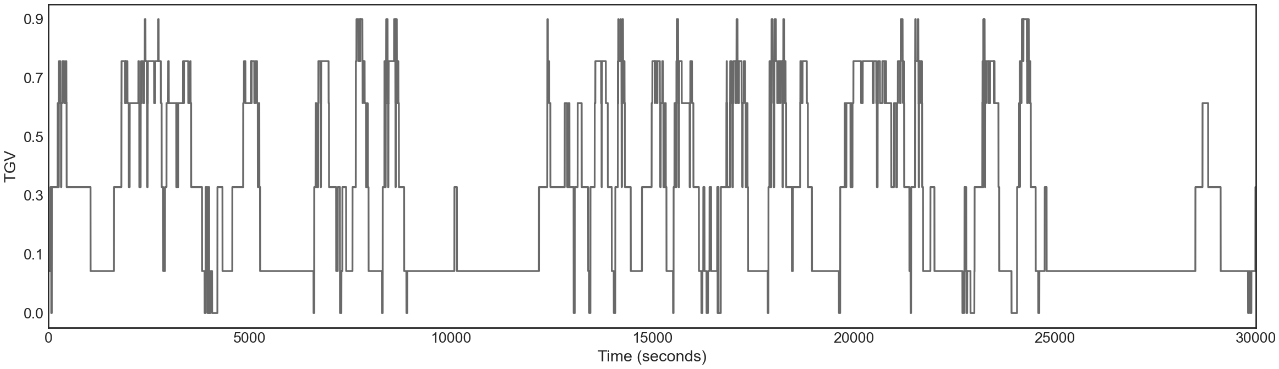


Figure 4.12

Sample TGV training data

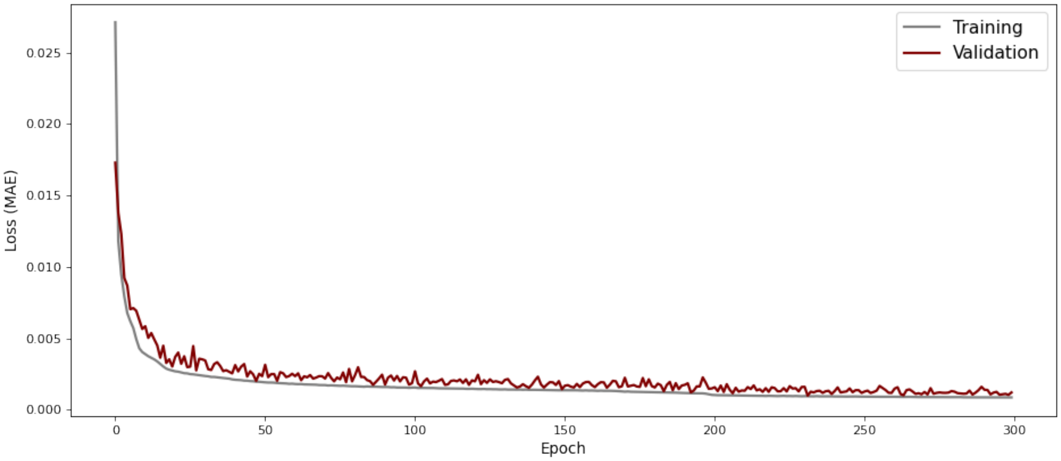


Figure 4.13

Operating vehicle data set: training and validation loss plots

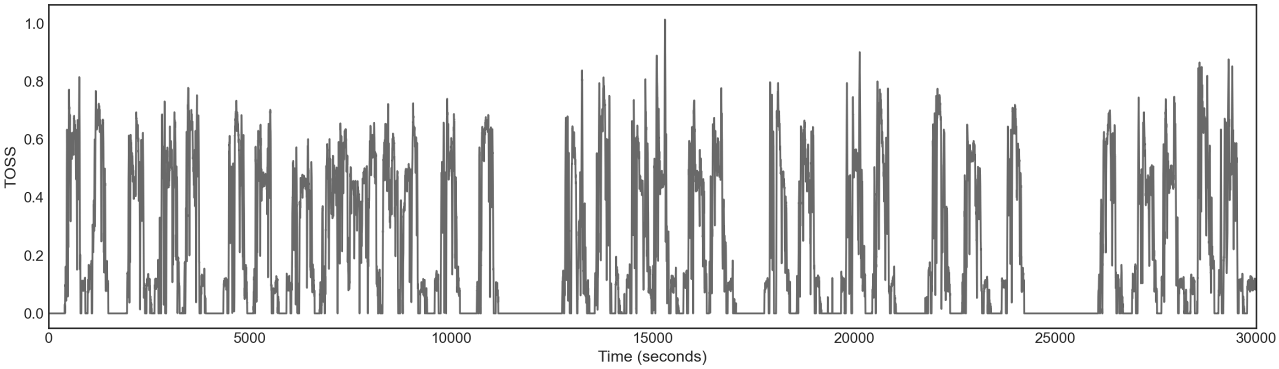


Figure 4.14

Sample TOSS test data before maintenance

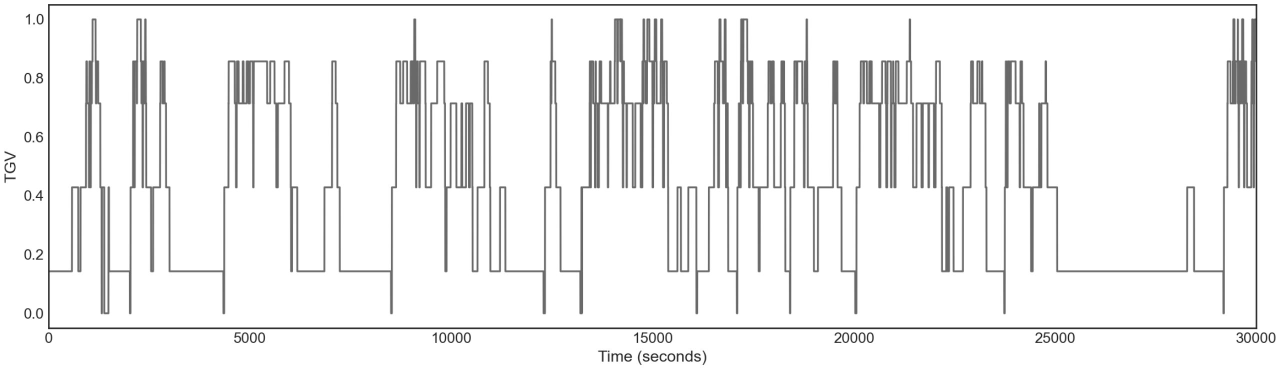


Figure 4.15

Sample TGV test data before maintenance

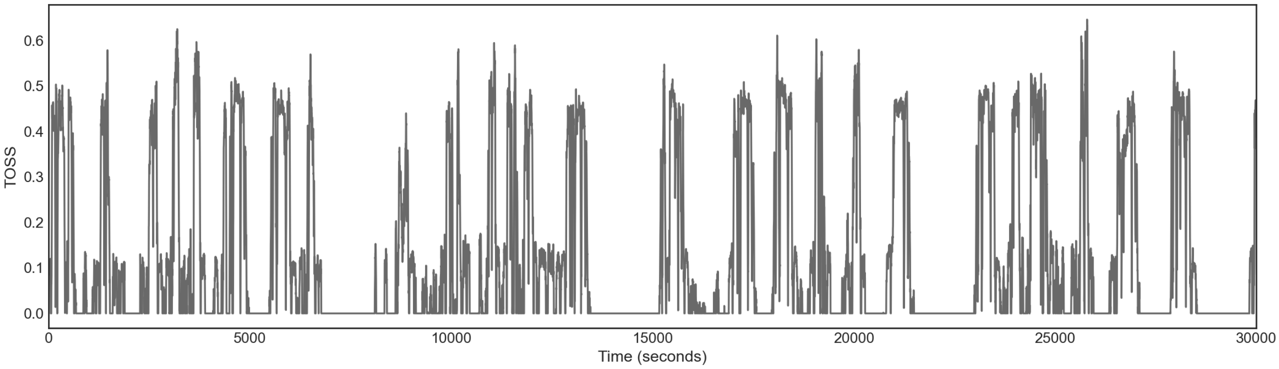


Figure 4.16

Sample TOSS test data after maintenance

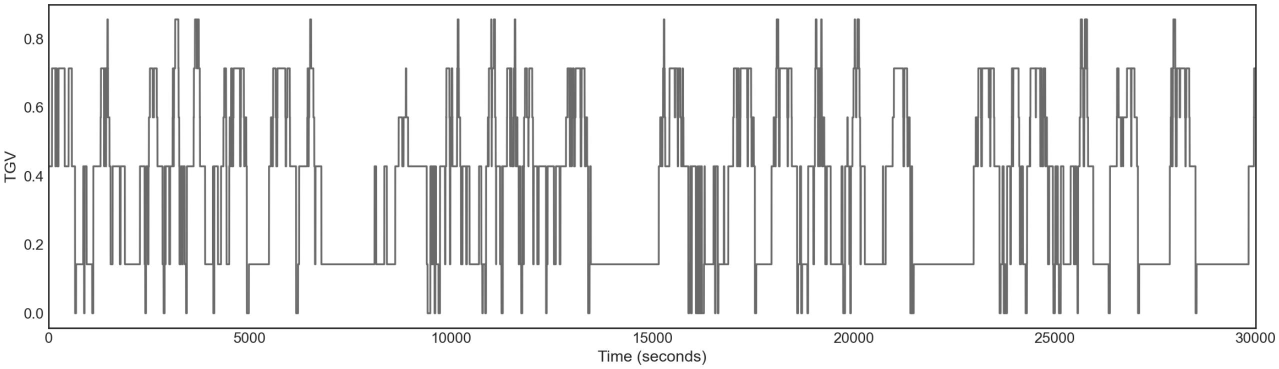


Figure 4.17

Sample TGV test data after maintenance

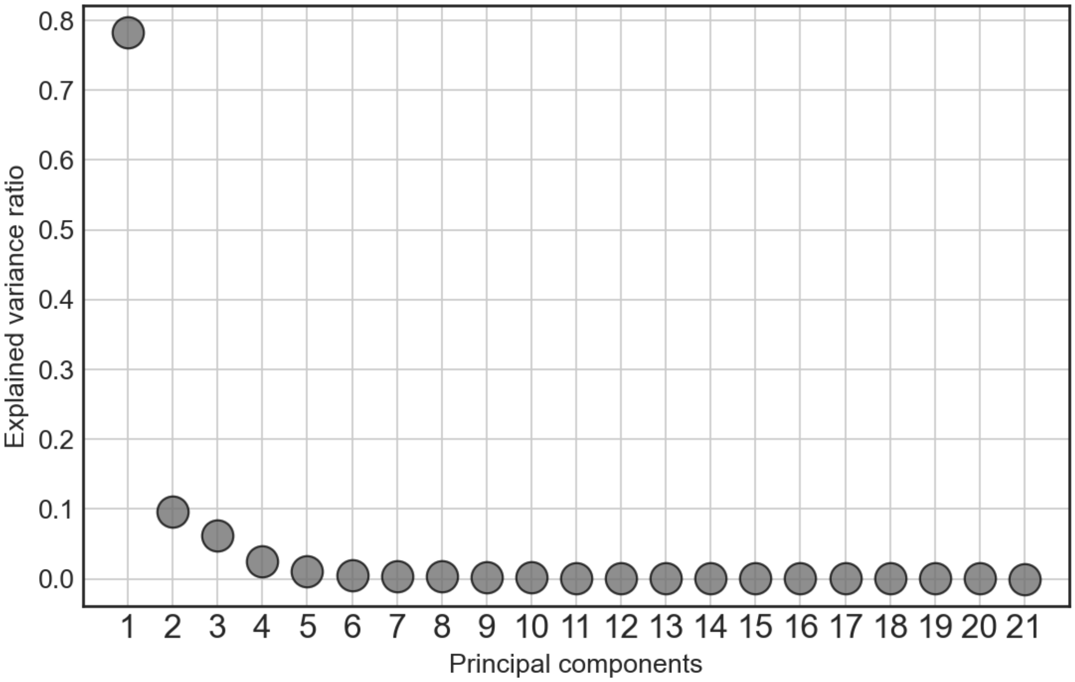


Figure 4.18

Explained variance ratio by PCA in operating vehicle test errors

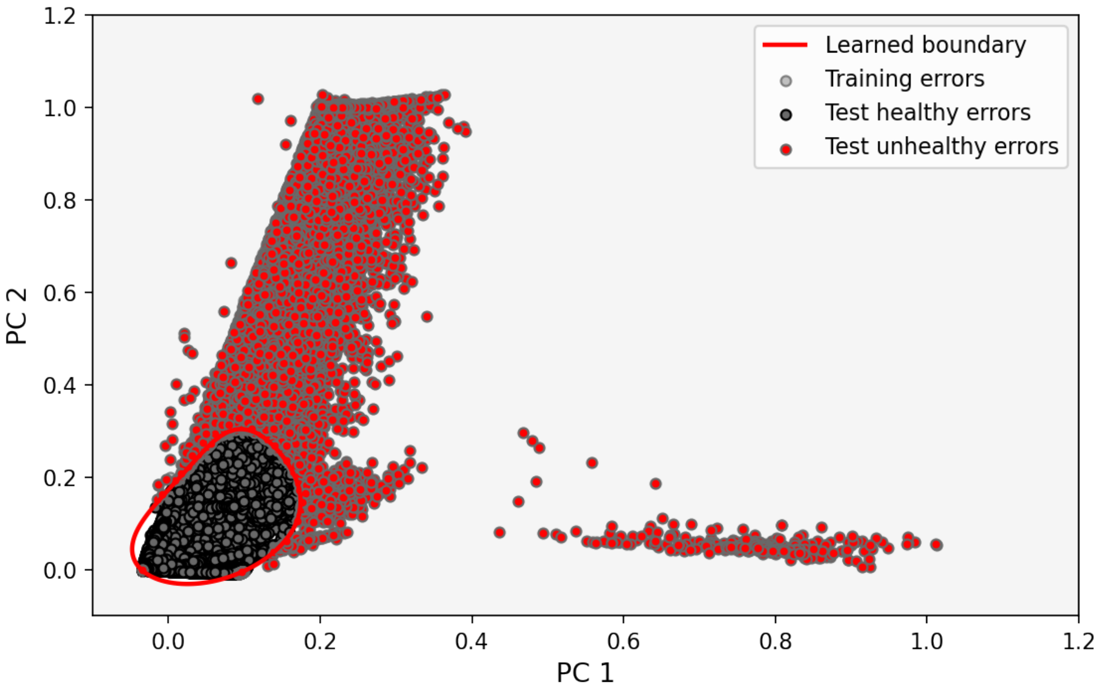


Figure 4.19

Healthy/unhealthy illustration of operating vehicle test data

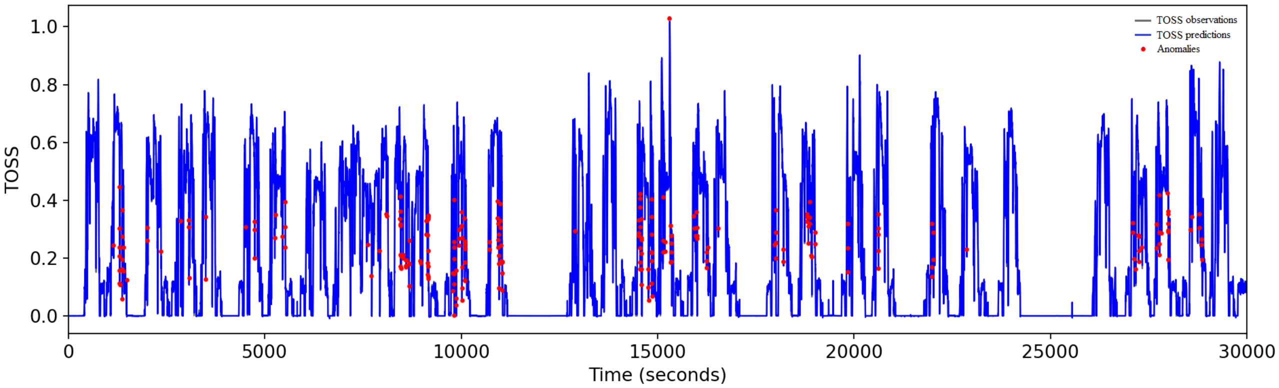


Figure 4.20

Sample TOSS test data prediction/anomaly before maintenance

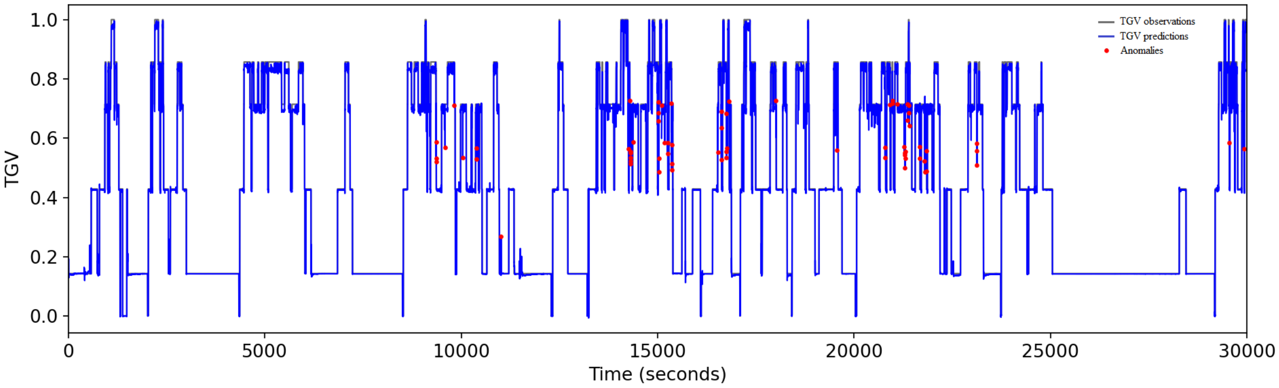


Figure 4.21

Sample TGV test data prediction/anomaly before maintenance

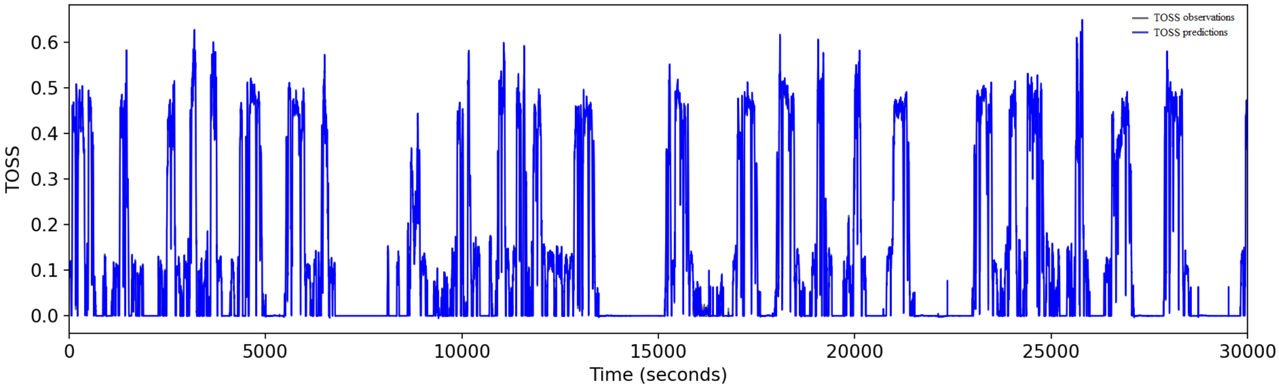


Figure 4.22

Sample TOSS test data prediction after maintenance

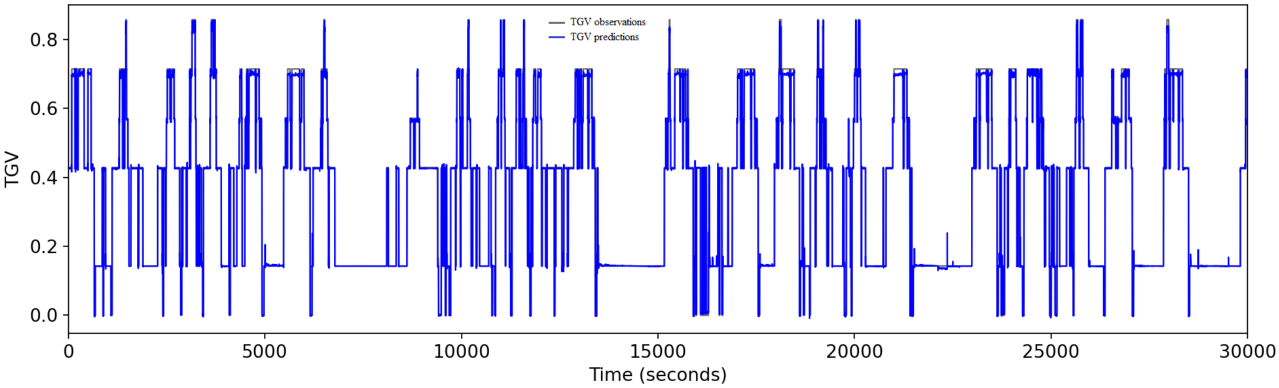


Figure 4.23

Sample TGV test data prediction after maintenance

CHAPTER V

CONCLUSION AND FUTURE RESEARCH

5.1 Chapter Structure

This chapter accomplishes the disseration by outlining the research summary and pointing the future research directions.

5.2 Research Summary

With the growing complexity of modern vehicles system, monitoring the behaviors of different subsystems of the vehicles becomes vital for decision makers to ensure the safety of the passengers, controlling the maintenance costs, and retaining the vehicle's life cycle. Nowadays, the modern vehicles comprise a variety of complex subsystems that are working together or individually to achieve an expected product. These complex components may cause faults and breakdowns in the vehicle that result in expensive maintenance costs, short life cycle of the vehicle, or even passengers casualty. Fault in the vehicle subsystems can be defined as an unexpected event that occurs at a certain timestamp and may lead to a worse event or even a series of other unexpected events. Therefore, it is highly significant to be aware of the imminent failures in the performance of various subsystems of the vehicle to make the best possible countermeasure to avoid extra losses.

Faults in the performance of the subsystems can be identified early as abnormal behavior a long time before emerging as critical issues. That is due to the fact that the vehicle performance begins degrading once a subsystem failure manifests. In fact, multiple occurrences of failures in the performance of the vehicle's components not only signal abnormality but also emphasize the necessity of deploying an appropriate maintenance. Thus, early detection of abnormal behaviors in the subsystems of the operating vehicle ensures a sufficient time interval to implement the required countermeasures that prevent extra losses, sustain the operating performance of the vehicle, and retain the passengers safety.

This dissertation in multivariate time series forecasting was accompanied by the fields of deep learning and statistical learning of big data; specially, operating vehicle's behavior prediction and unhealthy states detection are now a challenging problem in the modern vehicle with complex subsystems. This dissertation proposes novel single and multivariate models for effectively addressing the major drawbacks in predicting the behavior of operating vehicle and detecting unhealthy states as demonstrated here:

- Behavior prediction and abnormality detection in individual data channels using statistical (i.e., Auto Regressive Integrated Moving Average) and deep learning (i.e., Wavelet Neural Network) methods.
- Comparative study of series hybrid approaches to model and predict the vehicle operating states using single variate time series data channels
- Multivariate time series model for predicting vehicle behavior and detecting unhealthy states by integrating Long Short Term Memory (LSTM), Autoencoder scheme, and One-Class Support Vector Machine (OCSVM)

5.3 Future Research

The proposed dissertation developed novel contributions to monitor the operational behavior of operating vehicle using statistical and deep learning approaches. Besides, there are some future research revenues listed as follows:

1. **Convolutional LSTM network for prediction and anomaly detection in vehicle's multi-channel time series sequences.**

The objective of this study is to accelerate the learning process in the multi-channel data sequences using convolutional operations. To accomplish this goal, the proposed multi-layer LSTM autoencoder model can be improved by equipping the network weights with convolutional filters. These filters fasten the training phase by rapid extraction of meaningful information from the input data sequences.

2. **Transfer learning to address the time-varying challenges in the time series sequences of different subsystems.**

Next, transfer learning can be leveraged to address the time-varying challenges latent in the time series sequences of different subsystems of the operating vehicle. It will be helpful when the sufficient amount of training and testing data coming from the same distribution are not available. It also can be useful to share the learning knowledge obtained from modeling the operational behaviors of different subsystems of the vehicle of this research to model and predict the behavior of the other vehicles of the same family.

3. **Evaluating the performance of Bidirectional Encoder Representations from Transformers (BERT) for prediction and anomaly detection in the vehicle multi-channel operational data.**

BERT is a machine learning technique for bidirectional training of transformers that was developed by Google in 2018 for Natural Language Processing (NLP) [52]. This technique pre-trains a deep bidirectional model from data by jointly conditioning on both left and right context in all layers. The results showed that the bidirectionally training in the BERT model leads to a deeper sense of data comparing to the single-direction model. Due to the complexity of analyzing the multivariate time series of the operating vehicle, the applicability of the BERT technique for modeling and predicting the multivariate sequential data of the operating vehicle is worthy of investigation.

REFERENCES

[1] O. Abedinia, N. Amjady, and N. Ghadimi, “Solar energy forecasting based on hybrid neural network and improved metaheuristic algorithm,” *Computational Intelligence*, vol. 34, no. 1, 2018, pp. 241–260.

[2] C. C. Aggarwal, “An introduction to outlier analysis,” *Outlier analysis*, Springer, 2017, pp. 1–34.

[3] K. Aho, D. Derryberry, and T. Peterson, “Model Selection for Ecologists: The Worldviews of AIC and BIC,” *Ecology*, vol. 95, no. 3, 2014, pp. 631–636.

[4] H. N. Akouemo and R. J. Povinelli, “Probabilistic anomaly detection in natural gas time series data,” *International Journal of Forecasting*, vol. 32, no. 3, 2016, pp. 948–956.

[5] C. H. Aladag, E. Egrioglu, and C. Kadilar, “Forecasting nonlinear time series with a hybrid methodology,” *Applied Mathematics Letters*, vol. 22, no. 9, 2009, pp. 1467–1470.

[6] V. Alarcon-Aquino, J. M. Ramirez-Cortes, P. Gomez-Gil, O. Starostenko, and Y. Garcia-Gonzalez, “Network Intrusion Detection Using Self-Recurrent Wavelet Neural Network with Multidimensional Radial Wavelons,” *Information Technology and Control*, vol. 43, no. 4, 2014, pp. 347–358.

[7] A. K. Alexandridis and A. D. Zapranis, “Wavelet neural networks: A practical guide,” *Neural Networks*, vol. 42, 2013, pp. 1–27.

[8] M. Alizadeh, M. Hamilton, P. Jones, J. Ma, and R. Jaradat, “Vehicle operating state anomaly detection and results virtual reality interpretation,” *Expert Systems with Applications*, vol. 177, 2021, p. 114928.

[9] T. Amarbayasgalan, B. Jargalsaikhan, and K. H. Ryu, “Unsupervised novelty detection using deep autoencoders with density based clustering,” *Applied Sciences*, vol. 8, no. 9, 2018, p. 1468.

[10] M. Amer, M. Goldstein, and S. Abdennadher, “Enhancing one-class support vector machines for unsupervised anomaly detection,” *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, 2013, pp. 8–15.

[11] T. Andrysiak, Ł. Saganowski, M. Maszewski, and A. Marchewka, "Detection of Network Attacks Using Hybrid ARIMA-GARCH Model," *Advances in Dependability Engineering of Complex Systems*, Springer, 2017, pp. 1–12.

[12] G. Aradhye, A. Rao, and M. M. Mohammed, "A Novel Hybrid Approach for Time Series Data Forecasting Using Moving Average Filter and ARIMA-SVM," *Emerging Technologies in Data Mining and Information Security*, Springer, 2019, pp. 369–381.

[13] P. Areekul, T. Senjyu, H. Toyama, and A. Yona, "Notice of violation of ieee publication principles: A hybrid arima and neural network model for short-term price forecasting in deregulated market," *IEEE Transactions on Power Systems*, vol. 25, no. 1, 2009, pp. 524–530.

[14] S. Babbar and S. Chawla, "On Bayesian Network and Outlier Detection.," *COMAD*, 2010, p. 125.

[15] C. N. Babu and B. E. Reddy, "A moving-average filter based hybrid ARIMA–ANN model for forecasting time series data," *Applied Soft Computing*, vol. 23, 2014, pp. 27–38.

[16] R. M. Balabin and E. I. Lomakina, "Support Vector Machine Regression (SVR/LS-SVM)—an Alternative to Neural Networks (ANN) for Analytical Chemistry? Comparison of Nonlinear Methods on near Infrared (NIR) Spectroscopy Data," *Analyst*, vol. 136, no. 8, 2011, pp. 1703–1712.

[17] X. Baril, O. Coustié, J. Mothe, and O. Teste, "Application Performance Anomaly Detection with LSTM on Temporal Irregularities in Logs," *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1961–1964.

[18] G. A. Barreto, "Time Series Prediction with the Self-Organizing Map: A Review," *Perspectives of Neural-Symbolic Integration*, Springer, 2007, pp. 135–158.

[19] F. Battaglia and M. K. Protopapas, "Time-varying Multi-regime Models Fitting by Genetic Algorithms," *Journal of Time Series Analysis*, vol. 32, no. 3, 2011, pp. 237–252.

[20] Y. Becerikli, Y. Oysal, and A. F. Konar, "On a dynamic wavelet network and its modeling application," *Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003*, Springer, 2003, pp. 710–718.

[21] R. Benkercha and S. Moulahoum, "Fault detection and diagnosis based on C4. 5 decision tree algorithm for grid connected PV system," *Solar Energy*, vol. 173, 2018, pp. 610–634.

[22] P. Bereziński, B. Jasiul, and M. Szpyrka, "An Entropy-Based Network Anomaly Detection Method," *Entropy*, vol. 17, no. 4, 2015, pp. 2367–2408.

[23] A. M. Bianco, M. Garcia Ben, E. Martinez, and V. J. Yohai, "Outlier detection in regression models with arima errors using robust estimates," *Journal of Forecasting*, vol. 20, no. 8, 2001, pp. 565–579.

[24] S. A. Billings and H.-L. Wei, "A new class of wavelet networks for nonlinear system identification," *IEEE Transactions on neural networks*, vol. 16, no. 4, 2005, pp. 862–874.

[25] H. Boubaker, G. Canarella, R. Gupta, and S. M. Miller, "Hybrid ARFIMA wavelet artificial neural network model for DJIA Index Forecasting," *Available at SSRN 3660943*, 2020.

[26] G. E. Box and G. M. Jenkins, "Time Series Analysis: Forecasting and Control Holden-Day," *San Francisco*, 1970, p. 498.

[27] G. E. Box and G. C. Tiao, "Intervention analysis with applications to economic and environmental problems," *Journal of the American Statistical association*, vol. 70, no. 349, 1975, pp. 70–79.

[28] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: identifying density-based local outliers," *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[29] L. Cao and Q. Gu, "Dynamic Support Vector Machines for Non-Stationary Time Series Forecasting," *Intelligent Data Analysis*, vol. 6, no. 1, 2002, pp. 67–83.

[30] L. Cao, Y. Hong, H. Fang, and G. He, "Predicting Chaotic Time Series with Wavelet Networks," *Physica D: Nonlinear Phenomena*, vol. 85, no. 1-2, 1995, pp. 225–238.

[31] Y. Cao, J. Cao, Z. Zhou, and Z. Liu, "Aircraft Track Anomaly Detection Based on MOD-Bi-LSTM," *Electronics*, vol. 10, no. 9, 2021, p. 1007.

[32] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[33] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *arXiv preprint arXiv:1802.06360*, 2018.

[34] L. Chao, J. Zhipeng, and Z. Yuanjie, "A Novel Reconstructed Training-Set SVM with Roulette Cooperative Coevolution for Financial Time Series Classification," *Expert Systems with Applications*, vol. 123, 2019, pp. 283–298.

[35] S. Chauhan and L. Vig, "Anomaly detection in ECG time signals via deep long short-term memory networks," *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 2015, pp. 1–7.

[36] A. Chawla, P. Jacob, B. Lee, and S. Fallon, "Bidirectional LSTM autoencoder for sequence based anomaly detection in cyber security.," *International Journal of Simulation–Systems, Science & Technology*, 2019.

[37] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, 2018, pp. 1–12.

[38] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with ARIMA-GARCH model," *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 607–612.

[39] J. Chen, D. Pi, Z. Wu, X. Zhao, Y. Pan, and Q. Zhang, "Imbalanced satellite telemetry data anomaly detection model based on Bayesian LSTM," *Acta Astronautica*, vol. 180, 2021, pp. 232–242.

[40] K.-Y. Chen and C.-H. Wang, "A hybrid SARIMA and support vector machines in forecasting the production values of the machinery industry in Taiwan," *Expert Systems with Applications*, vol. 32, no. 1, 2007, pp. 254–264.

[41] Y. Chen, B. Yang, and J. Dong, "Time-Series Prediction Using a Local Linear Wavelet Neural Network," *Neurocomputing*, vol. 69, no. 4-6, 2006, pp. 449–465.

[42] Z. Chen, C. K. Yeo, B. S. Lee, and C. T. Lau, "Autoencoder-based network anomaly detection," *2018 Wireless Telecommunications Symposium (WTS)*. IEEE, 2018, pp. 1–5.

[43] C. Cheng, A. Sa-Ngasoongsong, O. Beyca, T. Le, H. Yang, Z. Kong, and S. T. Bukkapatnam, "Time Series Forecasting for Nonlinear and Non-Stationary Processes: A Review and Comparative Study," *Iie Transactions*, vol. 47, no. 10, 2015, pp. 1053–1071.

[44] F. Cheng, X. Guo, Y. Qi, J. Xu, W. Qiu, Z. Zhang, W. Zhang, and N. Qi, "Research on Satellite Power Anomaly Detection Method Based on LSTM," *2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA)*. IEEE, 2021, pp. 706–710.

[45] P. Cristea, R. Tuduce, and A. Cristea, "Time Series Prediction with Wavelet Neural Networks," *Proceedings of the 5th Seminar on Neural Network Applications in Electrical Engineering. NEUREL 2000 (IEEE Cat. No. 00EX287)*. 2000, pp. 5–10, IEEE.

[46] C. de Groot and D. Würtz, "Analysis of Univariate Time Series with Connectionist Nets: A Case Study of Two Classical Examples," *Neurocomputing*, vol. 3, no. 4, 1991, pp. 177–192.

[47] M. De Nadai and M. van Someren, "Short-Term Anomaly Detection in Gas Consumption through ARIMA and Artificial Neural Network Forecast," *2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) Proceedings*. 2015, pp. 250–255, IEEE.

[48] J. F. L. de Oliveira, L. D. S. Pacífico, P. S. G. de Mattos Neto, E. F. S. Barreiros, C. M. de Oliveira Rodrigues, and A. T. de Almeida Filho, "A hybrid optimized error correction system for time series forecasting," *Applied Soft Computing*, vol. 87, 2020, p. 105970.

[49] A. Delmas, M. Sallak, W. Schön, and L. Zhao, "Remaining Useful Life Estimation Methods for Predictive Maintenance Models: Defining Intervals and Strategies for Incomplete Data," *in Industrial Maintenance and Reliability Manchester, UK 12-15 June, 2018*, 2018, p. 48.

[50] K. Deng, X. Zhang, Y. Cheng, Z. Zheng, F. Jiang, W. Liu, and J. Peng, “A remaining useful life prediction method with long-short term feature processing for aircraft engines,” *Applied Soft Computing*, vol. 93, 2020, p. 106344.

[51] Derek Farren, Thai Pham, and Marco Alban-Hidalgo, “Low Latency Anomaly Detection and Bayesian Network Prediction of Anomaly Likelihood,”, 2016.

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

[53] L. A. Díaz-Robles, J. C. Ortega, J. S. Fu, G. D. Reed, J. C. Chow, J. G. Watson, and J. A. Moncada-Herrera, “A hybrid ARIMA and artificial neural networks model to forecast particulate matter in urban areas: The case of Temuco, Chile,” *Atmospheric Environment*, vol. 42, no. 35, 2008, pp. 8331–8340.

[54] S. d. O. Domingos, J. F. de Oliveira, and P. S. de Mattos Neto, “An intelligent hybridization of ARIMA with machine learning models for time series forecasting,” *Knowledge-Based Systems*, vol. 175, 2019, pp. 72–86.

[55] B. Doucoure, K. Agbossou, and A. Cardenas, “Time series prediction using artificial wavelet neural network and multi-resolution analysis: Application to wind speed data,” *Renewable Energy*, vol. 92, 2016, pp. 202–211.

[56] A. L. Ellefsen, V. Æsøy, S. Ushakov, and H. Zhang, “A Comprehensive Survey of Prognostics and Health Management Based on Deep Learning for Autonomous Ships,” *IEEE Transactions on Reliability*, vol. 68, no. 2, 2019, pp. 720–740.

[57] A. L. Ellefsen, E. Bjørlykhaug, V. Æsøy, S. Ushakov, and H. Zhang, “Remaining useful life predictions for turbofan engine degradation using semi-supervised deep architecture,” *Reliability Engineering & System Safety*, vol. 183, 2019, pp. 240–251.

[58] J. Fan and Q. Yao, “Nonlinear Time Series: Nonparametric and Parametric Methods Springer-Verlag,” *New York*, 2003.

[59] M. N. French, W. F. Krajewski, and R. R. Cuykendall, “Rainfall Forecasting in Space and Time Using a Neural Network,” *Journal of hydrology*, vol. 137, no. 1-4, 1992, pp. 1–31.

[60] Z. Ghahramani and S. T. Roweis, “Learning Nonlinear Dynamical Systems Using an EM Algorithm,” *Advances in Neural Information Processing Systems*, 1999, pp. 431–437.

[61] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, “Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks,” *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. 2017, pp. 140–145, IEEE.

[62] M. Goldstein and A. Dengel, “Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm,” *KI-2012: Poster and Demo Track*, 2012, pp. 59–63.

[63] A. M. González, A. S. Roque, and J. García-González, “Modeling and Forecasting Electricity Prices with Input/Output Hidden Markov Models,” *IEEE Transactions on power systems*, vol. 20, no. 1, 2005, pp. 13–24.

[64] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT press, 2016.

[65] N. J. Gordon, D. J. Salmond, and A. F. Smith, “Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation,” *IEE Proceedings F (Radar and Signal Processing)*. 1993, vol. 140, pp. 107–113, IET.

[66] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, “LSTM: A search space odyssey,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, 2016, pp. 2222–2232.

[67] G. Grudnitski and L. Osburn, “Forecasting S&P and Gold Futures Prices: An Application of Neural Networks,” *Journal of Futures Markets*, vol. 13, no. 6, 1993, pp. 631–643.

[68] Y. Gu, A. McCallum, and D. Towsley, “Detecting Anomalies in Network Traffic Using Maximum Entropy Estimation,” *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, 2005, pp. 32–32.

[69] N. Gugulothu, V. TV, P. Malhotra, L. Vig, P. Agarwal, and G. Shroff, “Predicting Remaining Useful Life Using Time Series Embeddings Based on Recurrent Neural Networks,” *arXiv preprint arXiv:1709.01073*, 2017.

[70] Z. Hajirahimi and M. Khashei, “Hybrid structures in time series modeling and forecasting: A review,” *Engineering Applications of Artificial Intelligence*, vol. 86, 2019, pp. 83–106.

[71] Z. Hajirahimi and M. Khashei, “Sequence in Hybridization of Statistical and Intelligent Models in Time Series Forecasting,” *Neural Processing Letters*, 2020, pp. 1–21.

[72] F. O. Heimes, “Recurrent Neural Networks for Remaining Useful Life Estimation,” *2008 International Conference on Prognostics and Health Management*. 2008, pp. 1–6, IEEE.

[73] H. S. Hippert, C. E. Pedreira, and R. C. Souza, “Neural Networks for Short-Term Load Forecasting: A Review and Evaluation,” *IEEE Transactions on power systems*, vol. 16, no. 1, 2001, pp. 44–55.

[74] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural computation*, vol. 9, no. 8, 1997, pp. 1735–1780.

[75] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, 1997, pp. 1735–1780.

[76] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artificial intelligence review*, vol. 22, no. 2, 2004, pp. 85–126.

[77] Q. Hou, J. Leng, G. Ma, W. Liu, and Y. Cheng, “An adaptive hybrid model for short-term urban traffic flow prediction,” *Physica A: Statistical Mechanics and its Applications*, vol. 527, 2019, p. 121065.

[78] C.-S. Hsu and J.-R. Jiang, “Remaining Useful Life Estimation Using Long Short-Term Memory Deep Learning,” *2018 IEEE International Conference on Applied System Invention (ICASI)*. 2018, pp. 58–61, IEEE.

[79] M. Hu, Z. Ji, K. Yan, Y. Guo, X. Feng, J. Gong, X. Zhao, and L. Dong, “Detecting anomalies in time series data via a meta-feature based approach,” *IEEE Access*, vol. 6, 2018, pp. 27760–27776.

[80] C.-G. Huang, H.-Z. Huang, and Y.-F. Li, “A Bidirectional LSTM Prognostics Method Under Multiple Operational Conditions,” *IEEE Transactions on Industrial Electronics*, vol. 66, no. 11, 2019, pp. 8792–8802.

[81] H. Huang, G. F. List, T. Tang, A. Demers, and T. Wang, “Hybrid Traffic Flow Forecasting Model Based on MRA,” *2009 International Conference on Measuring Technology and Mechatronics Automation*. IEEE, 2009, vol. 3, pp. 222–225.

[82] W. Huang, J. Zhang, H. Sun, H. Ma, and Z. Cai, “An anomaly detection method based on normalized mutual information feature selection and quantum wavelet neural network,” *Wireless Personal Communications*, vol. 96, no. 2, 2017, pp. 2693–2713.

[83] A. Jain and A. M. Kumar, “Hybrid Neural Network Models for Hydrologic Time Series Forecasting,” *Applied Soft Computing*, vol. 7, no. 2, 2007, pp. 585–592.

[84] L. Jiao, J. Pan, and Y. Fang, “Multiwavelet Neural Network and Its Approximation Properties,” *IEEE Transactions on neural networks*, vol. 12, no. 5, 2001, pp. 1060–1066.

[85] W. Jin, A. K. Tung, J. Han, and W. Wang, “Ranking outliers using symmetric neighborhood relationship,” *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2006, pp. 577–593.

[86] F. Johansson and G. Falkman, “Detection of Vessel Anomalies-a Bayesian Network Approach,” *2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information*. 2007, pp. 395–400, IEEE.

[87] S. Kadambe and P. Srinivasan, “Adaptive Wavelets for Signal Classification and Compression,” *AEU-International Journal of Electronics and Communications*, vol. 60, no. 1, 2006, pp. 45–55.

[88] S. Kanarachos, S.-R. G. Christopoulos, A. Chroneos, and M. E. Fitzpatrick, “Detecting Anomalies in Time Series Data via a Deep Learning Algorithm Combining Wavelets, Neural Networks and Hilbert Transform,” *Expert Systems with Applications*, vol. 85, 2017, pp. 292–304.

[89] S. Kanarachos, S.-R. G. Christopoulos, A. Chroneos, and M. E. Fitzpatrick, "Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and Hilbert transform," *Expert Systems with Applications*, vol. 85, 2017, pp. 292–304.

[90] G. Kar, B. Asiroglu, and F. S. Bir, "Scotto: Real-Time Driver Behavior Scoring Using In-Vehicle Data," *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–5.

[91] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM fully convolutional networks for time series classification," *IEEE access*, vol. 6, 2017, pp. 1662–1669.

[92] M. Khashei and M. Bijari, "An artificial neural network (p, d, q) model for timeseries forecasting," *Expert Systems with applications*, vol. 37, no. 1, 2010, pp. 479–489.

[93] M. Khashei, M. Bijari, and G. A. R. Ardali, "Hybridization of autoregressive integrated moving average (ARIMA) with probabilistic neural networks (PNNs)," *Computers & Industrial Engineering*, vol. 63, no. 1, 2012, pp. 37–45.

[94] M. Khashei and Z. Hajirahimi, "A comparative study of series arima/mlp hybrid models for stock price forecasting," *Communications in Statistics-Simulation and Computation*, vol. 48, no. 9, 2019, pp. 2625–2640.

[95] M. Khashei, S. R. Hejazi, and M. Bijari, "A New Hybrid Artificial Neural Networks and Fuzzy Regression Model for Time Series Forecasting," *Fuzzy sets and systems*, vol. 159, no. 7, 2008, pp. 769–786.

[96] R. Khemchandani and S. Chandra, "Regularized Least Squares Fuzzy Support Vector Regression for Financial Time Series Forecasting," *Expert Systems with Applications*, vol. 36, no. 1, 2009, pp. 132–138.

[97] T.-Y. Kim and S.-B. Cho, "Web traffic anomaly detection using C-LSTM neural networks," *Expert Systems with Applications*, vol. 106, 2018, pp. 66–76.

[98] V. Kodogiannis and A. Lolis, "Forecasting Financial Time Series Using Neural Network and Fuzzy System-Based Techniques," *Neural computing & applications*, vol. 11, no. 2, 2002, pp. 90–102.

[99] N. Kohzadi, M. S. Boyd, B. Kermanshahi, and I. Kaastra, "A Comparison of Artificial Neural Network and Time Series Models for Forecasting Commodity Prices," *Neurocomputing*, vol. 10, no. 2, 1996, pp. 169–181.

[100] Z. Kong, A. Oztekin, O. F. Beyca, U. Phatak, S. T. Bukkapatnam, and R. Komanduri, "Process Performance Prediction for Chemical Mechanical Planarization (CMP) by Integration of Nonlinear Bayesian Analysis and Statistical Modeling," *IEEE Transactions on Semiconductor Manufacturing*, vol. 23, no. 2, 2010, pp. 316–327.

[101] V. B. Krishna, R. K. Iyer, and W. H. Sanders, "ARIMA-Based Modeling and Validation of Consumption Readings in Power Grids," *International Conference on Critical Information Infrastructures Security*. 2015, pp. 199–210, Springer.

[102] V. Krishnamurthy and G. G. Yin, "Recursive Algorithms for Estimation of Hidden Markov Models and Autoregressive Models with Markov Regime," *IEEE Transactions on Information Theory*, vol. 48, no. 2, 2002, pp. 458–476.

[103] C.-M. Kuan and T. Liu, "Forecasting Exchange Rates Using Feedforward and Recurrent Neural Networks," *Journal of applied econometrics*, vol. 10, no. 4, 1995, pp. 347–364.

[104] K. K. Lai, L. Yu, S. Wang, and H. Wei, "A Novel Nonlinear Neural Network Ensemble Model for Financial Time Series Forecasting," *International Conference on Computational Science*. 2006, pp. 790–793, Springer.

[105] A. Lapedes and R. Farber, *Nonlinear Signal Processing Using Neural Networks: Forecast and System Modeling. Los Alamos National Lab. Los Alamos*, Tech. Rep., NM. Technical report LAUR872662, 1987.

[106] K. Lau and Q. Wu, "Local Prediction of Non-Linear Time Series Using Support Vector Regression," *Pattern recognition*, vol. 41, no. 5, 2008, pp. 1539–1547.

[107] Y.-S. Lee and L.-I. Tong, "Forecasting time series using a methodology based on autoregressive integrated moving average and genetic programming," *Knowledge-Based Systems*, vol. 24, no. 1, 2011, pp. 66–72.

[108] L. Lei, "Wavelet neural network prediction method of stock price trend based on rough set attribute reduction," *Applied Soft Computing*, vol. 62, 2018, pp. 923–932.

[109] J. Li, W. Pedrycz, and I. Jamal, "Multivariate Time Series Anomaly Detection: A Framework of Hidden Markov Models," *Applied Soft Computing*, vol. 60, 2017, pp. 229–240.

[110] Z. Li, J. Li, Y. Wang, and K. Wang, "A deep learning approach for anomaly detection based on SAE and LSTM in mechanical equipment," *The International Journal of Advanced Manufacturing Technology*, vol. 103, no. 1, 2019, pp. 499–510.

[111] Y. Liao, L. Zhang, and C. Liu, "Uncertainty Prediction of Remaining Useful Life Using Long Short-Term Memory Network Based on Bootstrap Method," *2018 IEEE International Conference on Prognostics and Health Management (ICPHM)*. 2018, pp. 1–8, IEEE.

[112] B. Lindemann, N. Jazdi, and M. Weyrich, "Anomaly detection and prediction in discrete manufacturing based on cooperative LSTM networks," *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2020, pp. 1003–1010.

[113] M. Lineesh and C. J. John, "Analysis of Non-Stationary Time Series Using Wavelet Decomposition," *Nature and Science*, vol. 8, no. 1, 2010, pp. 53–59.

[114] B. Liu, Y. Xiao, L. Cao, Z. Hao, and F. Deng, "Svdd-based outlier detection on uncertain data," *Knowledge and information systems*, vol. 34, no. 3, 2013, pp. 597–618.

[115] F. Liu, H. Jiang, Y. M. Lee, J. Snowdon, and M. Bobker, "Statistical Modeling for Anomaly Detection, Forecasting and Root Cause Analysis of Energy Consumption for a Portfolio of Buildings," *12th International Conference of the International Building Performance Simulation Association*, 2011, pp. 2530–2537.

[116] L.-l. Liu and Y. Liu, "MQPSO based on wavelet neural network for network anomaly detection," *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, 2009, pp. 1–5.

[117] G. M. Ljung and G. E. Box, "On a Measure of Lack of Fit in Time Series Models," *Biometrika*, vol. 65, no. 2, 1978, pp. 297–303.

[118] D. Lowe and A. Webb, "Time Series Prediction by Adaptive Networks: A Dynamical Systems Perspective," *IEE Proceedings F (Radar and Signal Processing)*. 1991, vol. 138, pp. 17–24, IET.

[119] W. Lu and A. A. Ghorbani, "Network Anomaly Detection Based on Wavelet Analysis," *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 2008, pp. 1–16.

[120] S. Maleki, S. Maleki, and N. R. Jennings, "Unsupervised anomaly detection with LSTM autoencoders using statistical data-filtering," *Applied Soft Computing*, 2021, p. 107443.

[121] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," *arXiv preprint arXiv:1607.00148*, 2016.

[122] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," *Proceedings*. Presses universitaires de Louvain, 2015, vol. 89, pp. 89–94.

[123] A. Mallak and M. Fathi, "Sensor and Component Fault Detection and Diagnosis for Hydraulic Machinery Integrating LSTM Autoencoder Detector and Diagnostic Classifiers," *Sensors*, vol. 21, no. 2, 2021, p. 433.

[124] M. McAleer and M. C. Medeiros, "A Multiple Regime Smooth Transition Heterogeneous Autoregressive Model for Long Memory and Asymmetries," *Journal of Econometrics*, vol. 147, no. 1, 2008, pp. 104–119.

[125] J. L. McClelland, D. E. Rumelhart, P. R. Group, et al., "Parallel distributed processing," *Explorations in the Microstructure of Cognition*, vol. 2, 1986, pp. 216–271.

[126] H. Mekki, A. Mellit, and H. Salhi, "Artificial neural network-based modelling and fault detection of partial shaded photovoltaic modules," *Simulation Modelling Practice and Theory*, vol. 67, 2016, pp. 1–13.

[127] J. M. P. Menezes Jr and G. A. Barreto, "Long-Term Time Series Prediction with the NARX Network: An Empirical Evaluation," *Neurocomputing*, vol. 71, no. 16-18, 2008, pp. 3335–3343.

[128] C. Merkl and F. Holzapfel, "Application of non-parametric methods for integrity monitoring of flight control systems," *IFAC-PapersOnLine*, vol. 51, no. 15, 2018, pp. 975–980.

[129] H. Miao, B. Li, C. Sun, and J. Liu, "Joint Learning of Degradation Assessment and RUL Prediction for Aeroengines via Dual-Task Deep LSTM Networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, 2019, pp. 5023–5032.

[130] H. Z. Moayedi and M. Masnadi-Shirazi, "Arima model for network traffic prediction and anomaly detection," *2008 International Symposium on Information Technology*. IEEE, 2008, vol. 4, pp. 1–6.

[131] A.-G. Model, "Detection of Network Attacks Using Hybrid," *Dependability Problems and Complex Systems: Proceedings of the Twelfth International Conference on Dependability and Complex Systems DepCoS-RELCOMEX*, 2017, p. 1.

[132] S. A. Monfared and D. Enke, "Volatility forecasting using a hybrid GJR-GARCH neural network model," *Procedia Computer Science*, vol. 36, 2014, pp. 246–253.

[133] H. Moradkhani, S. Sorooshian, H. V. Gupta, and P. R. Houser, "Dual State–Parameter Estimation of Hydrological Models Using Ensemble Kalman Filter," *Advances in water resources*, vol. 28, no. 2, 2005, pp. 135–147.

[134] N. Mousazadeh, A, A. Aghaei, M, and M. Moradzadeh F, "Forecasting Stock Market Using Wavelet Transforms and Neural Networks and ARIMA (Case study of price index of Tehran Stock Exchange)," *International Journal of Applied Operational Research*, vol. 5, no. 3, 2015, pp. 31–40.

[135] P. Moyo and J. Brownjohn, "Detection of Anomalous Structural Behaviour Using Wavelet Analysis," *Mechanical Systems and Signal Processing*, vol. 16, no. 2-3, 2002, pp. 429–445.

[136] M. Müter and N. Asaj, "Entropy-Based Anomaly Detection for in-Vehicle Networks," *2011 IEEE Intelligent Vehicles Symposium (IV)*. 2011, pp. 1110–1115, IEEE.

[137] M. K. Nammous and K. Saeed, "Natural language processing: Speaker, language, and gender identification with LSTM," *Advanced Computing and Systems for Security*, Springer, 2019, pp. 143–156.

[138] P. C. Nayak, K. Sudheer, D. Rangan, and K. Ramasastri, "A Neuro-Fuzzy Computing Technique for Modeling Hydrological Time Series," *Journal of Hydrology*, vol. 291, no. 1-2, 2004, pp. 52–66.

[139] H. 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*, vol. 57, 2021, p. 102282.

[140] H. Nie, G. Liu, X. Liu, and Y. Wang, "Hybrid of ARIMA and SVMs for short-term load forecasting," *Energy Procedia*, vol. 16, 2012, pp. 1455–1460.

[141] S. Novakov, C.-H. Lung, I. Lambadaris, and N. Seddigh, "Studies in Applying PCA and Wavelet Algorithms for Network Traffic Anomaly Detection," *2013 IEEE 14th International Conference on High Performance Switching and Routing (HPSR)*. 2013, pp. 185–190, IEEE.

[142] Y. Oussar, I. Rivals, L. Personnaz, and G. Dreyfus, "Training wavelet networks for nonlinear dynamic input–output modeling," *Neurocomputing*, vol. 20, no. 1-3, 1998, pp. 173–188.

[143] J. Park and I. W. Sandberg, "Universal Approximation Using Radial-Basis-Function Networks," *Neural computation*, vol. 3, no. 2, 1991, pp. 246–257.

[144] S. Pauwels and T. Calders, "Extending Dynamic Bayesian Networks for Anomaly Detection in Complex Logs," *arXiv preprint arXiv:1805.07107*, 2018.

[145] E. H. Pena, S. Barbon, J. J. Rodrigues, and M. L. Proença, "Anomaly Detection Using Digital Signature of Network Segment with Adaptive ARIMA Model and Paraconsistent Logic," *2014 IEEE Symposium on Computers and Communications (ISCC)*. 2014, pp. 1–6, IEEE.

[146] E. H. Pena, S. Barbon, J. J. Rodrigues, and M. L. Proença, "Anomaly detection using digital signature of network segment with adaptive ARIMA model and Paraconsistent Logic," *2014 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2014, pp. 1–6.

[147] E. H. Pena, M. V. de Assis, and M. L. Proença, "Anomaly Detection Using Forecasting Methods Arima and Hwds," *2013 32nd International Conference of the Chilean Computer Science Society (SCCC)*. 2013, pp. 63–66, IEEE.

[148] E. H. Pena, M. V. de Assis, and M. L. Proença, "Anomaly detection using forecasting methods arima and hwds," *2013 32nd International Conference of the Chilean Computer Science Society (SCCC)*. IEEE, 2013, pp. 63–66.

[149] A. Petropoulos, S. P. Chatzis, and S. Xanthopoulos, "A Hidden Markov Model with Dependence Jumps for Predictive Modeling of Multidimensional Time-Series," *Information Sciences*, vol. 412, 2017, pp. 50–66.

[150] H. T. Pham and B.-S. Yang, "Estimation and forecasting of machine health condition using ARMA/GARCH model," *Mechanical Systems and Signal Processing*, vol. 24, no. 2, 2010, pp. 546–558.

[151] S. Pittner, S. V. Kamarthi, and Q. Gao, "Wavelet Networks for Sensor Signal Classification in Flank Wear Assessment," *Journal of Intelligent Manufacturing*, vol. 9, no. 4, 1998, pp. 315–322.

[152] S. Postalcioglu and Y. Becerikli, "Wavelet networks for nonlinear system modeling," *Neural Computing and Applications*, vol. 16, no. 4-5, 2007, pp. 433–441.

[153] X. Qi, Y. Liu, Q. Guo, S. Yu, and J. Yu, "Performance prediction of a shower cooling tower using wavelet neural network," *Applied Thermal Engineering*, vol. 108, 2016, pp. 475–485.

[154] J. Qiu, Q. Du, W. Wang, K. Yin, and L. Chen, "Short-Term Performance Metrics Forecasting for Virtual Machine to Support Anomaly Detection Using Hybrid ARIMA-WNN Model," *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. 2019, vol. 2, pp. 330–335, IEEE.

[155] J. Qiu, Q. Du, W. Wang, K. Yin, and L. Chen, "Short-Term Performance Metrics Forecasting for Virtual Machine to Support Anomaly Detection Using Hybrid ARIMA-WNN Model," *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*. IEEE, 2019, vol. 2, pp. 330–335.

[156] Q. Quan, C. Hong-Yi, and Z. Rui, "Entropy Based Method for Network Anomaly Detection," *2009 15th IEEE Pacific Rim International Symposium on Dependable Computing*. 2009, pp. 189–191, IEEE.

[157] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceedings of the IEEE*, vol. 77, no. 2, 1989, pp. 257–286.

[158] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 427–438.

[159] P. Rao, S. Bukkapatnam, O. Beyca, Z. J. Kong, and R. Komanduri, "Real-Time Identification of Incipient Surface Morphology Variations in Ultraprecision Machining Process," *Journal of Manufacturing Science and Engineering*, vol. 136, no. 2, 2014.

[160] M. Ray, A. Rai, V. Ramasubramanian, and K. Singh, "ARIMA-WNN hybrid model for forecasting wheat yield time-Series data," *J. Ind. Soc. Agric. Stat*, vol. 70, no. 1, 2016, pp. 63–70.

[161] R. Rosa, F. Gomide, and R. Ballini, "Evolving hybrid neural fuzzy network for system modeling and time series forecasting," *2013 12th International Conference on Machine Learning and Applications*. IEEE, 2013, vol. 2, pp. 378–383.

[162] R. Sallehuddin, S. M. Shamsuddin, and S. Z. M. Hashim, "Hybridization model of linear and nonlinear time series data for forecasting," *2008 Second Asia International Conference on Modelling & Simulation (AMS)*. IEEE, 2008, pp. 597–602.

[163] J. M. B. Sánchez, D. N. Lugilde, C. de Linares Fernández, C. D. de la Guardia, and F. A. Sánchez, “Forecasting Airborne Pollen Concentration Time Series with Neural and Neuro-Fuzzy Models,” *Expert Systems with Applications*, vol. 32, no. 4, 2007, pp. 1218–1225.

[164] M. Santhosh, C. Venkaiah, and D. V. Kumar, “Ensemble empirical mode decomposition based adaptive wavelet neural network method for wind speed prediction,” *Energy conversion and management*, vol. 168, 2018, pp. 482–493.

[165] J. Sanz, R. Perera, and C. Huerta, “Fault diagnosis of rotating machinery based on auto-associative neural networks and wavelet transforms,” *Journal of Sound and Vibration*, vol. 302, no. 4-5, 2007, pp. 981–999.

[166] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, “Estimating the support of a high-dimensional distribution,” *Neural computation*, vol. 13, no. 7, 2001, pp. 1443–1471.

[167] T. Schreiber, “Interdisciplinary Application of Nonlinear Time Series Methods,” *Physics reports*, vol. 308, no. 1, 1999, pp. 1–64.

[168] M. Schuster, “Paliwal, and K. Kuldip,“Bidirectional Recurrent Neural Networks,”,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, 1997, pp. 2673–2681.

[169] S. H. Seifi, W. Tian, H. Doude, M. A. Tschopp, and L. Bian, “Layer-Wise Modeling and Anomaly Detection for Laser-Based Additive Manufacturing,” *Journal of Manufacturing Science and Engineering*, vol. 141, no. 8, 2019.

[170] S. H. Seifi, W. Tian, A. Yadollahi, H. Doude, and L. Bian, “In-Situ Fatigue Prediction of Direct Laser Deposition Parts Based on Thermal Profile,” *ASME International Mechanical Engineering Congress and Exposition*. 2019, vol. 59384, p. V02BT02A037, American Society of Mechanical Engineers.

[171] M. Shafie-Khah, M. P. Moghaddam, and M. Sheikh-El-Eslami, “Price forecasting of day-ahead electricity markets using a hybrid forecast method,” *Energy Conversion and Management*, vol. 52, no. 5, 2011, pp. 2165–2169.

[172] Y. Shen, X. Wang, and J. Chen, “Wind power forecasting using multi-objective evolutionary algorithms for wavelet neural network-optimized prediction intervals,” *Applied Sciences*, vol. 8, no. 2, 2018, p. 185.

[173] B. Shi, P. Wang, J. Jiang, and R. Liu, “Applying high-frequency surrogate measurements and a wavelet-ANN model to provide early warnings of rapid surface water quality anomalies,” *Science of the Total Environment*, vol. 610, 2018, pp. 1390–1399.

[174] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “A comparison of ARIMA and LSTM in forecasting time series,” *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 1394–1401.

[175] A. J. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression," *Statistics and computing*, vol. 14, no. 3, 2004, pp. 199–222.

[176] M. Sölch, J. Bayer, M. Ludersdorfer, and P. van der Smagt, "Variational Inference for On-Line Anomaly Detection in High-Dimensional Time Series," *arXiv preprint arXiv:1602.07109*, 2016.

[177] S. Soltani, "On the Use of the Wavelet Decomposition for Time Series Prediction," *Neurocomputing*, vol. 48, no. 1-4, 2002, pp. 267–277.

[178] H. Sun, D. Yan, N. Zhao, and J. Zhou, "Empirical investigation on modeling solar radiation series with ARMA–GARCH models," *Energy Conversion and Management*, vol. 92, 2015, pp. 385–395.

[179] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2002, pp. 535–548.

[180] A. Tealab, "Time series forecasting using artificial neural networks methodologies: A systematic review," *Future Computing and Informatics Journal*, vol. 3, no. 2, 2018, pp. 334–340.

[181] T. Teräsvirta, "Specification, Estimation, and Evaluation of Smooth Transition Autoregressive Models," *Journal of the american Statistical association*, vol. 89, no. 425, 1994, pp. 208–218.

[182] Y. Tian, M. Mirzabagheri, S. M. H. Bamakan, H. Wang, and Q. Qu, "Ramp loss one-class support vector machine; A robust and effective approach to anomaly detection problems," *Neurocomputing*, vol. 310, 2018, pp. 223–235.

[183] G. C. Tiao and R. S. Tsay, "Some Advances in Non-linear and Adaptive Modelling in Time-series," *Journal of forecasting*, vol. 13, no. 2, 1994, pp. 109–131.

[184] H. Tong, *Non-Linear Time Series: A Dynamical System Approach*, Oxford University Press, Oxford, UK, 1990.

[185] T. M. Tran, X.-M. T. Le, H. T. Nguyen, and V.-N. Huynh, "A novel non-parametric method for time series classification based on k-Nearest Neighbors and Dynamic Time Warping Barycenter Averaging," *Engineering Applications of Artificial Intelligence*, vol. 78, 2019, pp. 173–185.

[186] T. Tron, Y. S. Resheff, M. Bazhmin, D. Weinshall, and A. Peled, "ARIMA-Based Motor Anomaly Detection in Schizophrenia Inpatients," *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. 2018, pp. 430–433, IEEE.

[187] T. Tron, Y. S. Resheff, M. Bazhmin, D. Weinshall, and A. Peled, "ARIMA-based motor anomaly detection in schizophrenia inpatients," *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*. IEEE, 2018, pp. 430–433.

[188] V. TV, P. Malhotra, L. Vig, and G. Shroff, "Data-Driven Prognostics with Predictive Uncertainty Estimation Using Ensemble of Deep Ordinal Regression Models," *arXiv preprint arXiv:1903.09795*, 2019.

[189] W. Ullah, A. Ullah, I. U. Haq, K. Muhammad, M. Sajjad, and S. W. Baik, "CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks," *Multimedia Tools and Applications*, 2020, pp. 1–17.

[190] B. Vahidi, N. Ghaffarzadeh, S. H. Hosseinian, and S. M. Ahadi, "An approach to detection of high impedance fault using discrete wavelet transform and artificial neural networks," *Simulation*, vol. 86, no. 4, 2010, pp. 203–215.

[191] T. Van Gestel, J. A. Suykens, D.-E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, and J. Vandewalle, "Financial Time Series Prediction Using Least Squares Support Vector Machines within the Evidence Framework," *IEEE Transactions on neural networks*, vol. 12, no. 4, 2001, pp. 809–821.

[192] E. A. Wan and R. Van Der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No. 00EX373)*. 2000, pp. 153–158, Ieee.

[193] M. Wan, Y. Song, Y. Jing, and J. Wang, "Function-aware anomaly detection based on wavelet neural network for industrial control communication," *Security and Communication Networks*, vol. 2018, 2018.

[194] J. Wang, G. Wen, S. Yang, and Y. Liu, "Remaining Useful Life Estimation in Prognostics Using Deep Bidirectional Lstm Neural Network," *2018 Prognostics and System Health Management Conference (PHM-Chongqing)*. 2018, pp. 1037–1042, IEEE.

[195] L. Wang, "Advanced Multivariate Time Series Forecasting Models," *Journal of Mathematics and Statistics*, vol. 14, no. 1, 2018, pp. 253–260.

[196] L. Wang, H. Zou, J. Su, L. Li, and S. Chaudhry, "An ARIMA-ANN hybrid model for time series forecasting," *Systems Research and Behavioral Science*, vol. 30, no. 3, 2013, pp. 244–259.

[197] S. Wang, X. Zhang, D. Gao, B. Chen, Y. Cheng, Y. Yang, W. Yu, Z. Huang, and J. Peng, "A Remaining Useful Life Prediction Model Based on Hybrid Long-Short Sequences for Engines," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2018, pp. 1757–1762, IEEE.

[198] Y. Wang, L. Han, W. Liu, S. Yang, and Y. Gao, "Study on wavelet neural network based anomaly detection in ocean observing data series," *Ocean Engineering*, vol. 186, 2019, p. 106129.

[199] Y. Wang, T. Zheng, Y. Zhao, J. Jiang, Y. Wang, L. Guo, and P. Wang, "Monthly water quality forecasting and uncertainty assessment via bootstrapped wavelet neural networks under missing data for Harbin, China," *Environmental Science and Pollution Research*, vol. 20, no. 12, 2013, pp. 8909–8923.

[200] Z. Wang, X. Liu, Y. Liu, J. Liang, and V. Vinciotti, "An Extended Kalman Filtering Approach to Modeling Nonlinear Dynamic Gene Regulatory Networks via Short Gene Expression Time Series," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 6, no. 3, 2009, pp. 410–419.

[201] K.-W. Wong and A. C.-S. Leung, "On-Line Successive Synthesis of Wavelet Networks," *Neural Processing Letters*, vol. 7, no. 2, 1998, pp. 91–100.

[202] D. Wu, Z. Jiang, X. Xie, X. Wei, W. Yu, and R. Li, "LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial IoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, 2019, pp. 5244–5253.

[203] Z. Wu, H. Luo, Y. Yang, X. Zhu, and X. Qiu, "An Unsupervised Degradation Estimation Framework for Diagnostics and Prognostics in Cyber-Physical System," *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*. 2018, pp. 784–789, IEEE.

[204] T. Xia, Y. Song, Y. Zheng, E. Pan, and L. Xi, "An ensemble framework based on convolutional bi-directional LSTM with multiple time windows for remaining useful life estimation," *Computers in Industry*, vol. 115, 2020, p. 103182.

[205] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," *arXiv preprint arXiv:1510.01553*, 2015.

[206] A. H. Yaacob, I. K. Tan, S. F. Chien, and H. K. Tan, "Arima based network anomaly detection," *2010 Second International Conference on Communication Software and Networks*. IEEE, 2010, pp. 205–209.

[207] P. K. Yadav, P. F. Pope, and K. Paudyal, "THRESHOLD AUTOREGRESSIVE MODELING IN FINANCE: THE PRICE DIFFERENCES OF EQUIVALENT ASSETS 1," *Mathematical Finance*, vol. 4, no. 2, 1994, pp. 205–221.

[208] N. Ye and Q. Chen, "An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems," *Quality and Reliability Engineering International*, vol. 17, no. 2, 2001, pp. 105–112.

[209] Q. Yu, L. Jibin, and L. Jiang, "An improved ARIMA-based traffic anomaly detection algorithm for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 12, no. 1, 2016, p. 9653230.

[210] Z. Yuan, W. Wang, H. Wang, and S. Mizzi, “Combination of cuckoo search and wavelet neural network for midterm building energy forecast,” *Energy*, 2020, p. 117728.

[211] D. Zeng, J. Xu, J. Gu, L. Liu, and G. Xu, “Short Term Traffic Flow Prediction Using Hybrid ARIMA and ANN Models,” *2008 Workshop on Power Electronics and Intelligent Transportation System*. 2008, pp. 621–625, IEEE.

[212] D. Zeng, J. Xu, J. Gu, L. Liu, and G. Xu, “Short term traffic flow prediction using hybrid ARIMA and ANN models,” *2008 Workshop on Power Electronics and Intelligent Transportation System*. IEEE, 2008, pp. 621–625.

[213] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlüter, and H. Ney, “A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition,” *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2462–2466.

[214] G. Zhang, B. E. Patuwo, and M. Y. Hu, “Forecasting with artificial neural networks:: The state of the art,” *International journal of forecasting*, vol. 14, no. 1, 1998, pp. 35–62.

[215] G. P. Zhang, “Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model,” *Neurocomputing*, vol. 50, 2003, pp. 159–175.

[216] G. P. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” *Neurocomputing*, vol. 50, 2003, pp. 159–175.

[217] G. P. Zhang, “Neural Networks for Time-Series Forecasting.,” 2012.

[218] G. P. Zhang and V. Berardi, “Time Series Forecasting with Neural Network Ensembles: An Application for Exchange Rate Prediction,” *Journal of the operational research society*, vol. 52, no. 6, 2001, pp. 652–664.

[219] J. Zhang, P. Wang, R. Yan, and R. X. Gao, “Long Short-Term Memory for Machine Remaining Life Prediction,” *Journal of manufacturing systems*, vol. 48, 2018, pp. 78–86.

[220] J. Zhang, Y.-M. Wei, D. Li, Z. Tan, and J. Zhou, “Short term electricity load forecasting using a hybrid model,” *Energy*, vol. 158, 2018, pp. 774–781.

[221] L. Zhang, W. Zhou, and L. Jiao, “Wavelet Support Vector Machine,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, 2004, pp. 34–39.

[222] Q. Zhang, “Regressor selection and wavelet network construction,” *Proceedings of 32nd IEEE Conference on Decision and Control*. IEEE, 1993, pp. 3688–3693.

[223] Q. Zhang, “Using wavelet network in nonparametric estimation,” *IEEE Transactions on Neural networks*, vol. 8, no. 2, 1997, pp. 227–236.

[224] Q. Zhang and A. Benveniste, “Wavelet networks,” *IEEE transactions on Neural Networks*, vol. 3, no. 6, 1992, pp. 889–898.

[225] Y. Zhang and G. Cao, "V-PADA: Vehicle-platoon-aware data access in VANETs," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 5, 2011, pp. 2326–2339.

[226] Y. Zhao, B. Deng, C. Shen, Y. Liu, H. Lu, and X.-S. Hua, "Spatio-temporal autoencoder for video anomaly detection," *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 1933–1941.

[227] Z. Zhao, C. Xu, and B. Li, "A LSTM-Based Anomaly Detection Model for Log Analysis," *Journal of Signal Processing Systems*, 2021, pp. 1–7.

[228] S. Zheng, K. Ristovski, A. Farahat, and C. Gupta, "Long Short-Term Memory Network for Remaining Useful Life Estimation," *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. 2017, pp. 88–95, IEEE.

[229] C. Zhou, K. Yin, Y. Cao, and B. Ahmed, "Application of Time Series Analysis and PSO–SVM Model in Predicting the Bazimen Landslide in the Three Gorges Reservoir, China," *Engineering geology*, vol. 204, 2016, pp. 108–120.

[230] M. Y. Zhou, J. T. Xi, and J. Q. Yan, "Adaptive Direct Slicing with Non-Uniform Cusp Heights for Rapid Prototyping," *The International Journal of Advanced Manufacturing Technology*, vol. 23, no. 1-2, 2004, pp. 20–27.

[231] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, "Variational LSTM enhanced anomaly detection for industrial big data," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, 2020, pp. 3469–3477.

[232] B. Zhu and S. Sastry, "Revisit dynamic arima based anomaly detection," *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*. IEEE, 2011, pp. 1263–1268.