Applications of Machine Learning Algorithms in Wireless Sensor Networks

by

Lei Yang

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Control Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

As a key infrastructure of constructing the Internet of Things, wireless sensor networks (WSNs) have attracted plenty of research interest. It is expected to be widely applied in almost every aspect of future life. Nowadays, some preliminary applications and prototypes have emerged in various fields from military applications to health and environmental applications, etc. However, many conceptual and practical problems are still required to be solved. In WSNs, some common research problems include sensor routing and clustering, data fusion, sensor localization, communication signal processing, intelligent event detection and decision making, etc. Recently, with the booming of cloud computing, Machine Learning (ML) based methods have arisen to provide many novel and effective solutions for a variety of problems in WSNs. The advantages of ML based methods are promising and can significantly boost the application and development of WSNs. In this thesis, three topics in WSNs are mainly studied. Firstly, the problem of redundant transmission reduction is studied and ML methods have been applied in the proposed prediction-based data fusion to reduce the number of wireless transmission. Secondly, the problem of communication channel equalization is studied, ML based equalization methods have been proposed and discussed. Finally, the problem of intelligent event detection is studied in the WSN-based fluid pipeline leak monitoring applications where Deep Learning (DL) and enhanced model based leak detection and localization methods are proposed and discussed.

Preface

The research works in Chapter 3 of the thesis is on data fusion based wireless transmission reduction. The research works in Chapter 4 is on communication channel equalization and signal detection. The research works in Chapters 5-7 are on the intelligent event detection and decision making.

- Chapter 3 has been published as "Conditional training based GM and GM-OPELM data fusion schemes in wireless sensor networks" in *2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM).*

- Chapter 4 has been published as "Channel equalization and detection with ELM-based regressors for OFDM systems" in *IEEE Communications Letters.*

- Chapter 5 has been published as "A Novel PPA method for fluid pipeline leak detection based on OPELM and bidirectional LSTM" in *IEEE Access.*

- Chapter 6 has been submitted for publication as "A BiLSTM based pipeline leak detection and disturbance assisted localization method" in *IEEE Sensors Journal.*

- Chapter 7 has been submitted for publication as "Combined dual-prediction based data fusion and enhanced leak detection and isolation method in WSN pipeline monitoring system" in *IEEE Transactions on Automation Science and Engineering.*

To my wife Dan Wang, my daughter Jiaqi Yang for their endless love.

Acknowledgements

First and foremost I am grateful to my esteemed supervisor Prof. Qing Zhao for her invaluable guidance, continuous support, and patience during my PhD study. I have benefited greatly from her wealth of knowledge. I am extremely grateful that she took me on as a student and continued to have faith in me over the years.

Also, I would like to thank Prof. Yindi Jing for her treasured technical support on my research works of data fusion and channel equalization. The wise advice and meticulous editing she has generously provided are really influential in shaping the papers for publication and the completion of the thesis.

I also want to give my appreciation to Prof. Mahdi Tavakoli, a member of my supervision committee, for his insightful comments and advice in the progression of my thesis.

Finally, I would like to express my gratitude to my wife Dan Wang, my daughter Jiaqi Yang and my parents. Without their tremendous understanding and encouragements, it would be impossible for me to complete my study.

Contents

List of Tables

List of Figures

List of Acronyms

ML	Machine Learning
DL	Deep Learning
WSN	Wireless Sensor Networks
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
KNN	k-Nearest Neighbors
SVM	Support Vector Machine
GAN	Generative Adversarial Network
PCA	Principal Component Analysis
PPA	Pressure Point Analysis
SOM	Self-Organizing Map
ELM	Extreme Learning Machine
NPW	Negative Pressure Wave
ACC	Accuracy
TPR	True Positive Rate
FDR	False Discovery Rate
LSTM	Long-Short Term Memory
MMSE	Minimum Mean Square Error
TDOA	Time Difference of Arrival
OPELM	Optimally-Pruned Extreme Learning Machine
BiLSTM	Bidirectional Long-Short Term Memory

Chapter 1

Introduction

1.1 Introduction of Wireless Sensor Networks

Nowadays, through the use of distributed sensors, almost every aspect of our life is interconnected as indicated in Fig. 1.1. By deploying networked devices, the entire physical world can be closely coupled with the assistance of communications and information technologies. In order to measure the sophisticated dynamic systems, wireless sensor networks (WSNs) formed by a large number of interconnected distributed sensors are established. In WSNs the sensed measurements and control instructions are transmitted via distributed sensors. By implementing a specific set of data processing mechanism, the key information needed can be mined and retrieved from large scales of data. After processing the acquired data with certain algorithms, one or more statistical models representing physical features of the investigated target can be obtained. Therefore, the model based prediction and detection can be performed to meet practical needs. WSNs are regarded as a revolutionary information gathering method to build future interconnected systems with great improvement in reliability and efficiency. Compared with the wired solution, WSNs feature easier deployment of devices and better flexibility. With the rapid technological development of sensors, WSNs will become the key technology for Internet of Things (IoTs) [1].

WSNs have great potential for building powerful applications as shown in Fig. 1.2. In military applications [2], WSNs can be deployed in battlefield to

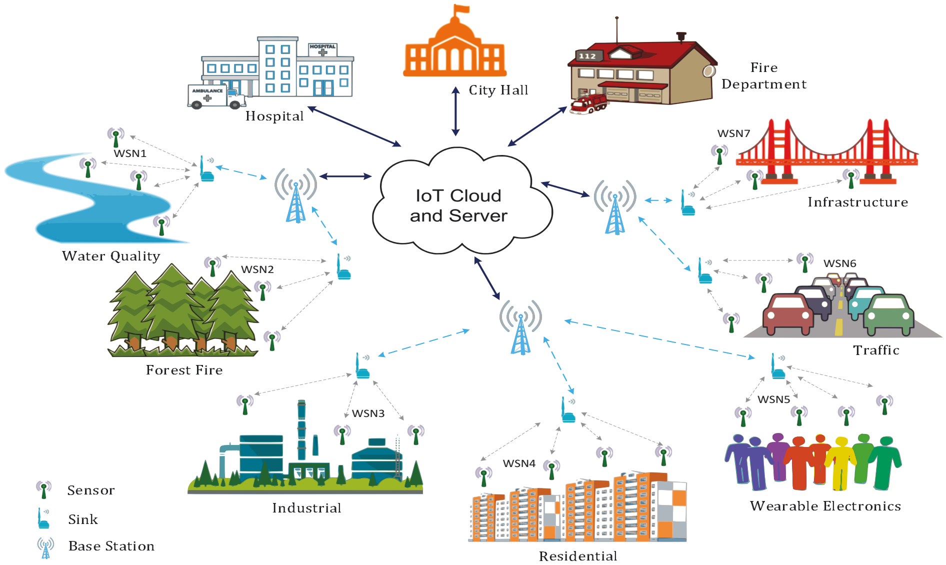

Figure 1.1: WSNs in Internet of Things.

implement opposing area surveillance, attack detection, vehicle and equipment monitoring, etc. In environmental applications [3], WSNs can be utilized for forest fire detection, landslide detection, agricultural environment monitoring, etc. In health applications [4], WSNs can build a smart system from patient information gathering to doctor's decision making and drug administration. In home applications [5], WSNs can help to realize home automation, indoor environment management and home security protection, etc. In commercial applications [6], WSNs can be applied in inventory and parking area management, vehicle tracking and theft prevention, etc. However, alongside the fast development of WSNs, some common challenges and issues have emerged and attracted tremendous research interests lately. Some of the research topics in WSNs are to be introduced in the next section.

1.2 Scope of Research

In WSNs, there are various topics [7, 8] that are studied such as sensor clustering and routing, energy efficiency, sensor deployment, sensor localization and object tracking, redundant data aggregation, communication, monitor-

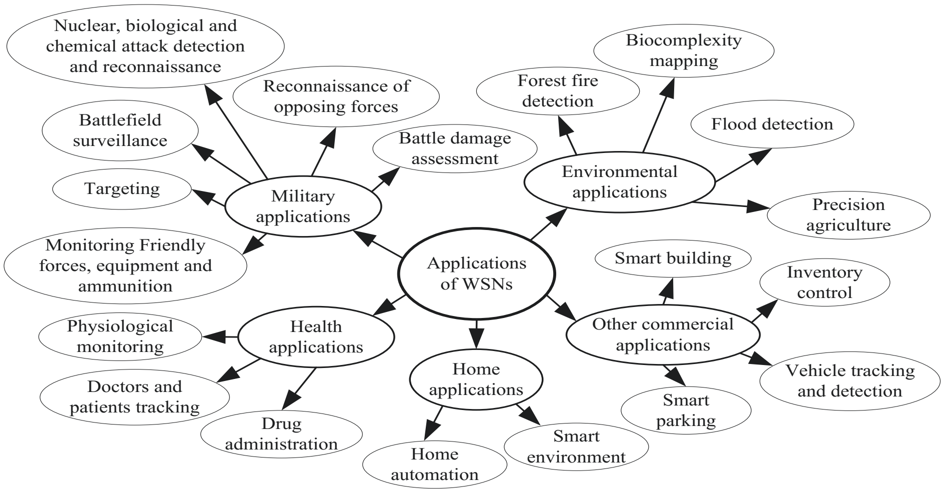

Figure 1.2: Applications of WSNs.

ing and event detection, security and intrusion prevention, etc. In this thesis, some topics that are mainly researched on include "redundant transmission reduction", "communication channel equalization and signal detection", "event detection and decision making". As shown in Fig. 1.3, a WSN-based pipeline leak monitoring application is taken as an example to introduce the investigated problems in this thesis. The details are as following.

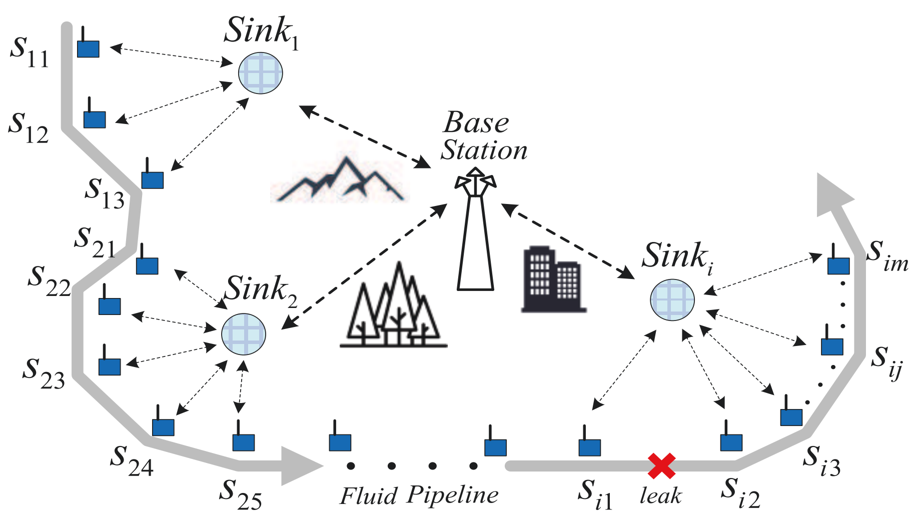

Figure 1.3: An example application scenario.

The pipeline can be stretched through a vast area with various complex landscapes, e.g., the mountain, forest and even metropolitan. Numerous sensor nodes S_{ij} are mounted along the pipeline. The sensor nodes are clustered into multiple groups where the data collected by each group is transmitted to its sink node, e.g., $Sink_i$. The sink nodes then continue transmitting the collections to the base station via long distance wireless transmission. The transmission environments are complex where the communication channels are usually interfered with by different types of barriers. After the measurements from multiple sensors are converged in the base station, a decision making algorithm is implemented to detect and locate the leak event. Since the pressure or flow readings are usually contaminated by noise or other interference, the detection reliability and localization accuracy can deteriorate. To keep the monitoring system working effectively, some problems are to be studied and solved as follows.

1.2.1 Redundant Transmission Reduction

In WSN systems, e.g., the pipeline monitoring system in Fig. 1.3, numerous low-cost pressure or flow sensors are installed along the pipeline. Usually, the cost-sensitive sensors are driven by battery or solar power, thus their power storage is very limited. Those sensors continuously measure pressure or flow readings and send the measurements to their sink nodes. However, wireless transmission is energy-consuming, and it may deplete the power in sensor nodes in the short term and paralyze the whole monitoring system. Therefore, the problem of reducing redundant and unnecessary wireless transmission becomes critical for prolonging the active lifetime of WSNs.

One of the approaches to reduce data transmission is the dual-prediction based data fusion method. The idea is straightforward. If the sink node, e.g., $Sink_i$, can accurately predict the values collected by its children nodes, e.g., $\{S_{i1}, \cdots, S_{im}\}$, the transmissions from the children nodes to sink node can be eliminated. Existing studies of prediction-based data fusion have been mainly focused on time-series prediction. They usually employ some commonly used models to implement the time-series prediction [9, 10], e.g., autoregressive (AR) model, moving average (MA) model, and autoregressive integrated moving average (ARIMA) model, autoregressive exogenous (ARX) model, etc. However, the prediction accuracy is not satisfactory and the models lack flexibility to accommodate the dynamic and varying environment. On the other hand, ML methods have the advantages of superior approximation ability for complex and dynamic models. They can generate more accurate time-series prediction and further decrease the number of data transmissions.

Especially, with the emergence of deep learning (DL) based time-series forecasting methods such as LSTM, the prediction accuracy is greatly improved. But the high computational complexity of DL-based methods may hinder the practical application. In this thesis, some research and case studies are implemented, and the ML based data fusion methods are proposed and discussed. The details are to be introduced in the Chapter 3 and part of Chapter 7.

1.2.2 Channel Equalization and Signal Detection

In WSNs, the communications between a sink node, e.g., $Sink_i$, and the base station can be treated as point-to-point communications. The local collections within a cluster are transmitted from sink nodes to the base station. The sink nodes and base station are usually equipped with continuous power supply which allows them to implement high speed wireless transmissions and complex calculations, etc. However, the wireless communication channels are usually interfered with by the landscape, plants and buildings as shown in Fig. 1.3. The blocks and barriers may result in channel distortion and multipath fading where the channels become frequency selective. Consequently, it will cause the inter-symbol interference where symbols received over the direct or the shortest reflecting paths interfere with previous symbols arriving at the same time over longer delayed paths. To reduce the frequency selectivity, channel equalization methods are implemented.

Commonly used channel equalizers include minimum mean squared error (MMSE) equalizer, zero forcing equalizer, and adaptive equalizer. MMSE equalizer is a linear equalizer designed to minimize the mean squared error between the received and transmitted signals. Zero forcing equalizer approximates the inverse of the communication channel with a linear filter. Adaptive equalizer updates filter coefficients in the process of equalization. These traditional equalization methods often have very high computational complexity when the number of taps of the channel impulse response (CIR) is high [11]. ML based methods have been widely applied in channel equalization [12], whose complexity is mainly determined by the number of training symbols and the size of the employed approximation structure, but does not depend on the CIR. Also, the learning and adaptive processing capabilities make ML-based equalizer more effective in dynamic and time-varying communication environments [13]. In Chapter 4, a ML based method is proposed for channel equalization, which is shown to outperform several existing methods on both the computational efficiency and communication performance.

1.2.3 Event Detection in WSNs Monitoring System

The WSNs monitoring system has been applied in many fields for event detection, such as environmental disaster detection [14–16], water quality monitoring [17], and water pipeline leak detection [18], etc. In WSNs monitoring systems, multiple sensors are deployed in designated areas to sense required information. The acquired information then converges to the base station for decision making. However, the data analysis based detection methods may often result in false or missing alarms because of the strong background noise or low signal quality. For example, in the water pipeline leak monitoring system as shown in Fig. 1.3, the frequently appeared pressure and flow fluctuations may be misidentified as leaks. The high false alarm rate will deteriorate the reliability of the monitoring system and result in unnecessary labor costs. On the other hand, the missing alarm may cause disastrous consequences with the huge property and environmental loss. In this thesis, the topic of "event detection in WSNs" is studied in the context of pipeline leak monitoring applications, where the ML-based and model-based leak detection and localization schemes are developed in Chapters 5, 6, 7.

1.3 Pros and Cons of Applying ML in WSN

ML was introduced in the late 1950s as a technique for artificial intelligence (AI) [19]. It is a process that automatically improves or learns from the study or experience, and acts without being explicitly programmed. The following two classical definitions capture the essence of ML:

- The development of computer models for learning processes that provide solutions to the problem of knowledge acquisition and enhance the performance of developed systems [20].
- The adoption of computational methods for improving machine performance by detecting and describing consistencies and patterns in training data [21].

In the last decades, ML techniques have been used extensively in a wide range of tasks including classification, regression and density estimation in various areas such as bioinformatics, speech recognition, spam detection, computer vision, fraud detection and communication networks etc. The algorithms and techniques used come from many diverse fields including statistics, mathematics, neuroscience, and computer science.

The algorithms in the ML family can usually be categorized into supervised and unsupervised learning, where "supervised/unsupervised" indicates whether the data are well-labeled. Semi-supervised learning refers to the case where the database is partially labeled, and it can be seen as the combination of supervised and unsupervised cases. Later, reinforcement learning emerged as a new category that was inspired by behavioral psychology. It is concerned with an scheme of reward/punishment, which is connected to its environment via perception and action. Recently, deep learning has been extensively developed and exploited. It is a subcategory of the supervised learning which utilizes complex neural networks structure to implement the model learning. The taxonomy of main ML algorithms is shown in Fig. 1.4. Most algorithms have found applications in WSNs.

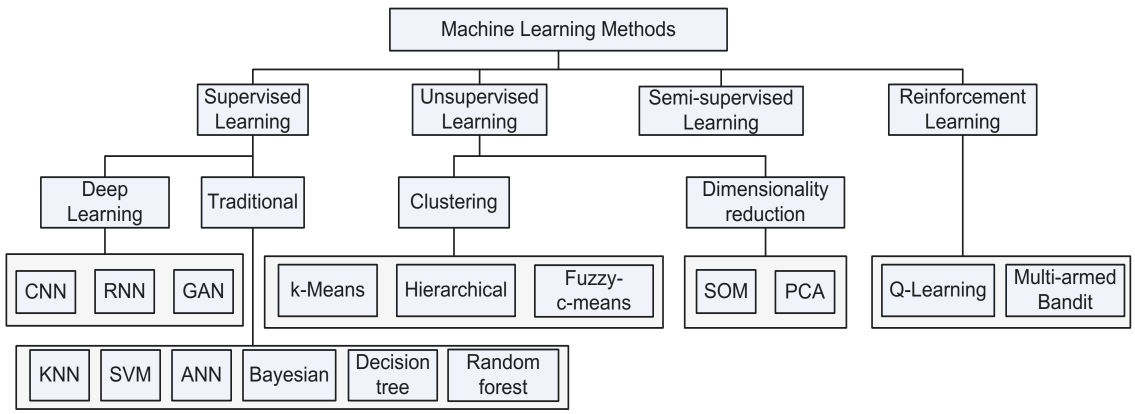

Figure 1.4: Category of Machine Learning method.

Tab. 1.1 lists some of the research works of applying ML methods in WSNs since 2017. It can be seen that the application of ML methods in WSNs has become a research hot spot, and with the development of more advanced ML techniques, the research trend will keep uprising. Generally speaking, the importance of ML in WSNs applications is mainly due to the following reasons [22].

- The online model updating ability of ML methods make it a prior solution in modeling dynamic environment. For example, in WSNs based landscape monitoring system, the network topology can be varying due to sensor failure or sensor relocation. ML methods can dynamically rebuild the connection model and choose optimal clustering and routing scheme.

- In most of the applications, it can be difficult to develop precise mathematical models to approximate the environment of interest. ML methods

Table 1.1: Recent publications on applying ML in WSNs.

	Routing and MAC protocol	Clustering and Data aggregation	Localization and Object targeting	Fault and Security
Bayesian Statistics	[23][24]	[25],[26]	[27],[28]	[29],[30]
K-Nearest Neighbors	[31] [32]	[33],[34]	[35],[36]	[37]
Support Vector Machine	[38]	[39, 40]	[41], [42],[43]	[44],[45]
Artificial Neural Networks	[46], [47]	[48],[49]	[50],[51],[52]	[53],[54]
Extreme Learning Machine	[55] [56]	[57],[58]	[59],[60]	[61] [62]
Self-Organizing map	[63],[64]	[65],[66]	[67]	[68],[69]
Principal Component	[70]	[71] [72] [73]	[74],[75]	[76],[77]
K-Means	[78] [79]	[80],[81]	[82] [83]	[84]
Reinforcement Learning	[85] [86]	[87]		[88]
Deep Learning	[89] [90]	[91] [92]	[93] [94]	[95] [96]

provide effective solutions that make the establishment of low-complexity, good approximation model feasible.

- The strong temporal and spatial correlation exploitation ability makes ML methods good choices in event detection, fault node tolerance and prediction based data fusion.

- The strong online learning ability of ML methods helps to upgrade its intelligence by learning more data sets, which can improve the reliability and accuracy of decision making. For example, in WSNs based intrusion detection systems, the detection accuracy gets improved by learning from experiences over time.

- WSNs may be used for collecting new knowledge about unreachable, dangerous locations (e.g., volcano eruption and waste water monitoring) in exploratory applications. Due to the unexpected behavior patterns that may arise in such scenarios, system designers may develop solutions that initially may not operate as expected, but the robust ML algorithms are able to calibrate itself to the newly acquired knowledge and improve the model adaptability.

On the other hand, there are a few drawbacks and limitations that should be considered when using ML techniques in WSNs.

- ML techniques drain a considerable percentage of energy to implement model learning. This is undesirable for resource limited systems such as WSNs. Also, the requirement of computational capability may not be available in the cost-sensitive sensor nodes.

- Most of the ML methods, especially the newly developed DL methods such as convolutional neural networks (CNNs) and long-short term memory (LSTM), etc., usually need a long and extensive training process to generate a model with good approximation and generality. This may not be applicable in some time-sensitive monitoring systems.

- The high computational complexity of most ML algorithms makes them hard to be embedded in distributed calculation systems. Most of the learning processes in such systems are implemented in the resource-capable computational units. Therefore, the lower-tier learning and decision making are unfeasible, causing the data transmission amount to increase.

- Compared to traditional statistical model-based methods, ML methods usually demand a large number of data samples to construct a model with good generality. This may require the algorithm designer to have a prior knowledge of the data formation process.

1.4 Outline and Contributions

In this thesis, some ML based methods particularly the extreme learning machine (ELM) and LSTM have been applied to solve some common issues in WSNs systems. The remainder of the thesis and the research contributions are summarized as follows.

- In Chapter 2, as the main ML algorithms adopted in this thesis, the preliminaries of ELM and LSTM algorithms are briefly reviewed.

- In Chapter 3, the problem of redundant transmission reduction is studied and a prediction-based data fusion method is proposed. It can greatly decrease the number of sensed data transmissions by improving prediction accuracy. Moreover, the training energy and time spent on model learning are also significantly decreased, thus, the method prolongs the active lifetime of sensor nodes.

- In Chapter 4, the problem of communication channel equalization is studied, where a combined multiple ELM regressors based channel equalization and detection structure is proposed for strong frequency-selective channels. The proposed methods greatly improve the detection performance compared with several existing ELM based equalization methods. Moreover, the complexity of the proposed methods is lower than existing methods and the advantage further outstands as the modulation order increases.

- In Chapter 5, WSNs based event detection is investigated in a pipeline leak detection scenario. An effective pressure point analysis (PPA) leak detection method based on optimally-pruned extreme learning machine (OPELM) combining bidirectional LSTM (BiLSTM) is proposed, which achieves higher detection accuracy and significant less false alarms than existing ML based methods. Performance of the proposed method is assessed and compared with various ML based methods through multiple experiments on different industrial data sets.

- In Chapter 6, the problem of pipeline leak detection and localization is studied in WSNs monitoring system. Firstly, the strong pattern classification ability of BiLSTM is utilized to detect not only leak impacts but also other non-leak pressure disturbances. It achieves high detection accuracy with rare false alarms. Then, the non-leak pressure disturbances under normal operations are utilized to perform online negative pressure wave (NPW) speed estimation. The estimated NPW speed is further employed in leak localization, hence the accuracy is greatly improved than the conventional approach with constant NPW speed.

- In Chapter 7, the problems of transmission reduction and leak detection are further investigated in a pipeline network monitoring system. An ELM based data fusion method and an enhanced leak detection and localization method are proposed. The performance of the proposed data fusion method is compared with other ML based methods and it achieves better transmission reduction efficacy. Also, the proposed leak detection and localization method outperforms existing model based methods.

- Chapter 8 concludes the thesis and introduces future research works.

Chapter 2

Preliminaries

2.1 Extreme Learning Machine

The ELM was proposed by Huang et al. in [97]. It employs the single hidden layer feed-forward neural network (SLFN) structure and can achieve an extremely fast learning speed. A unique and essential property of ELM compared to the conventional neural network is that it does not need to tune the parameters of the hidden layer. As proposed and proved in [97], the parameters of hidden nodes (input weights and biases) are randomly assigned and remain fixed for calculating the output layer weights afterwards. The output weights are analytically determined by the minimum least-squares solutions of a general system of linear equations. In [98], ELM with randomly generated hidden nodes has been substantiated to have faster learning speed with similar or much better generalization performance than traditional support vector machine (SVM) and least-squares SVM (LSSVM). Generally speaking, the ELM operates as an universal approximator, which can produce superior approximation performance in most cases and can learn thousands of times faster than other conventional algorithms. However, due to the structural simplicity, ELM may has some disadvantages in processing high dimensional and large scale data set. For example, in the fields of natural language processing and image processing, the performance of ELM is not comparable with the deep learning based neural networks. In the following subsections, the principle of ELM and its variants are introduced.

2.1.1 Basic Extreme Learning Machine

ELM works in the structure of SLFNs where the hidden layer parameters need not to be iteratively adjusted. The main feature of ELM is that all the hidden node parameters including input weights and biases are randomly generated even before training when the activation function in the hidden layer is infinitely differentiable. These parameters remain fixed after their generation. Notably, the learning process of the basic ELM consists of two steps. First, the input vectors are randomly mapped into a feature space associated with the hidden layer output matrix. Then, the standard optimization method is used to find the solution that minimizes the training errors. The major difference among supervised, semi-supervised and unsupervised ELMs lies in how to solve for the output weights.

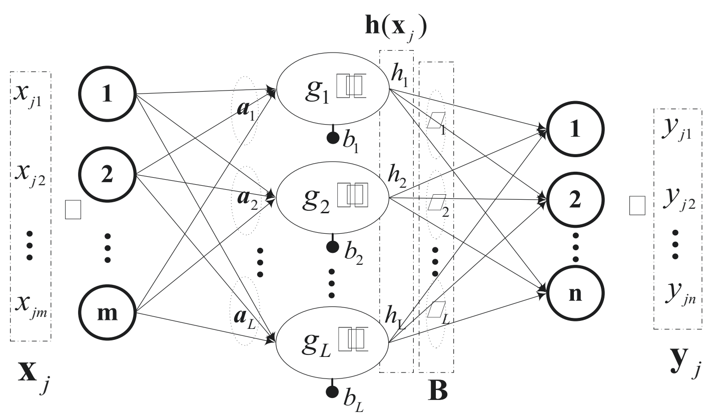

Figure 2.1: General structure of basic ELM.

First, the output function of an SLFN with L hidden nodes can be written by

$$\mathbf{y}_j = \sum_{i=1}^{L} \boldsymbol{\beta}_i g_i(\mathbf{a}_i, b_i, \mathbf{x}_j) = \mathbf{h}(\mathbf{x}_j)\mathbf{B}, \tag{2.1}$$

where $\boldsymbol{\beta}_i \in R^n, \mathbf{a}_i, \mathbf{x}_j \in R^m, b_i \in R, j = 1, \cdots, N$ and N denotes the number of data samples. $\mathbf{h}(\mathbf{x}_j) = [h_1(\mathbf{x}_j), \cdots, h_L(\mathbf{x}_j)]$ is the output row vector of the hidden layer with respect to the input vector $\mathbf{x}_j$. Also, $\mathbf{B} = [\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_L]^T$ is a $L \times n$ output weights matrix between the hidden layer and the output layer,

$\boldsymbol{\beta}_i = [\beta_i^1, \cdots, \beta_i^n]^T$ is the output weight vector between the output nodes and the i-th neuron in the hidden layer. $g_i(\mathbf{a}_i, b_i, \mathbf{x}_j)$ is the activation function.

If a supervised learning problem is considered, for a given training set of $(\mathbf{X}, \mathbf{Y}) = (\mathbf{x}_j, \mathbf{y}_j)_{j=1}^N$, from Eqn. (2.1), the N equations above can be expressed compactly as

$$\mathbf{HB} = \mathbf{Y}, \tag{2.2}$$

where

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \mathbf{h}(\mathbf{x}_2) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} g(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & g(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ g(\mathbf{a}_1, b_1, \mathbf{x}_N) & \cdots & g(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}_{N \times L}, \tag{2.3}$$

$$\mathbf{Y} = \left[\mathbf{y}_1^T, \cdots, \mathbf{y}_N^T\right]^T.$$

Thus, when the training data are inputs to the SLFN, $\mathbf{H}$ can be obtained using Eqn. (2.3). The only unknown parameters are the output weights matrix $\mathbf{B}$ between the hidden layer and the output layer. Once $\mathbf{H}$ is determined, the relationship between the hidden layer and the output layer is linear, and the output weighting matrix $\mathbf{B}$ can be analytically solved by solving a linear estimation problem at a much faster speed than the traditional gradient descent tuning process. A least-squares (LS) solution $\mathbf{B}^*$ of the linear Eqn. (2.2) is as follows

$$\mathbf{B}^* = \arg\min_{\mathbf{B}} \|\mathbf{Y} - \mathbf{HB}\|^2, \tag{2.4}$$

where $\|\cdot\|$ is a norm in Euclidean space or l_2-norm. Then, the minimum norm LS solution of Eqn. (2.4) is unique according to Theorem 5.1 in [97], which is

$$\mathbf{B}^* = \mathbf{H}^\dagger \mathbf{Y}. \tag{2.5}$$

where $\mathbf{H}^\dagger$ is the Moore-Penrose generalized inverse of matrix $\mathbf{H}$. The Moore-Penrose generalized inverse of a matrix can be written as $\mathbf{H}^\dagger = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ for the case when $N > L$ with nonsingular $\mathbf{H}^T\mathbf{H}$, whereas $\mathbf{H}^\dagger = \mathbf{H}^T(\mathbf{H}\mathbf{H}^T)^{-1}$ for the case when $N < L$ with a nonsingular $\mathbf{H}\mathbf{H}^T$, respectively. Usually, the number of hidden nodes is less than that of training samples, i.e., $N > L$, then Eqn. (2.5) can be rewritten as

$$\mathbf{B}^{*} = (\mathbf{H}^{T}\mathbf{H})^{-1}\mathbf{H}^{T}\mathbf{Y}. \tag{2.6}$$

Notably, the SVD method can be generally employed to calculate the Moore-Penrose generalized inverse whether $\mathbf{H}^{T}\mathbf{H}$ is nonsingular or not. Hence, based on the analysis above, the main merits of the ELM method are summarized as follows:

- The major advantage of ELM is its extremely fast learning speed compared to the other regular learning algorithms;
- It can achieve better generalization and scalability performance than SVM-based methods [98];
- It can perform efficient and effective parallel computing during the process of training and testing, similarly to most neural networks.

However, the basic ELM algorithm still has some disadvantages as follows.

- In ELM, overfitting may happen when the noise or random fluctuations in the training data is captured and learned as features by the model;
- The generalization performance of basic ELM may deteriorate since only the empirical risk related training error is minimized in the optimization of Eqn. (2.4), whereas the structural risk embodied by the norm of weights is not considered.
- It is not robust to the extreme data or outliers in the training set and becomes worse even than the other regularized learning algorithms.

In order to resolve the above issues of a basic ELM, several improved versions of ELM such as regularized extreme learning machine (RELM) [98, 99], weighted regularized ELM (WRELM) [99], and outlier-robust ELM (ORELM) [101] have been proposed.

2.1.2 Regularized Extreme Learning Machine

According to Bartlett's theory [102], not only the training error but also the norm of weights are required to be minimized simultaneously so as to address the over-fitting problem and improve the generalization ability for feedforward neural networks. The RELM is proposed where the minimization principle has to be applied for both empirical risk and structural risk. The principal can be expressed as

$$\mathbf{B}^* = \arg\min_{\mathbf{B}} \mathcal{C}\|\mathbf{Y} - \mathbf{H}\mathbf{B}\|^2 + \|\mathbf{B}\|^2, \tag{2.7}$$

where $\mathcal{C}$ denotes a regularization parameter which represents the proportion between the training error and the norm of output weights and can provide a tradeoff between them. Obviously, the optimization problem in Eqn. (2.7) without constraints is equivalent to the following minimization problem with an equality constraint

$$\begin{aligned} \mathbf{B}^* \;&=\; \arg\min_{\mathbf{B}} \mathcal{C}\|\xi\|^2 + \|\mathbf{B}\|^2 \\ &\text{s.t. } \mathbf{Y} - \mathbf{H}\mathbf{B} = \xi, \end{aligned} \tag{2.8}$$

or equivalently,

$$\begin{aligned} \mathbf{B}^* \;&=\; \arg\min_{\mathbf{B}} \|\xi\|^2 + \mathcal{C}\|\mathbf{B}\|^2 \\ &\text{s.t. } \mathbf{Y} - \mathbf{H}\mathbf{B} = \xi, \end{aligned} \tag{2.9}$$

The corresponding Lagrangian function for Eqn. (2.8) is defined as

$$\mathcal{L}(\mathbf{B}, \xi, \lambda) = \mathcal{C}\|\boldsymbol{\xi}\|^2 + \|\mathbf{B}\|^2 + \lambda^T(\mathbf{Y} - \mathbf{H}\mathbf{B} - \boldsymbol{\xi}), \tag{2.10}$$

where $\boldsymbol{\lambda} = [\lambda_1, \cdots, \lambda_N]^T$ and $\boldsymbol{\xi} = [\xi_1, \cdots, \xi_N]^T$ denote the column vector of the Lagrangian multipliers and the N training errors respectively. By applying Karush-Kuhn-Tucker (KKT) conditions, the optimal solution can be expressed as

$$\mathbf{B}^* = \begin{cases} (\mathbf{H}^T\mathbf{H} + \frac{\mathbf{I}}{\mathcal{C}})^{-1}\mathbf{H}^T\mathbf{Y}, & N > L, \\ \\ \mathbf{H}^T(\mathbf{H}\mathbf{H}^T + \frac{\mathbf{I}}{\mathcal{C}})^{-1}\mathbf{Y}, & N < L. \end{cases} \tag{2.11}$$

According to the ridge regression theory [103], the optimal solution to give in Eqn. (2.11) is stabler and has better generalization ability than Eqn. (2.6) without regularization.

2.1.3 Weighted Regularized Extreme Learning Machine

In the practical situations encountered, outliers may be present in the process of data acquisition and lead to a high training error. Thus, the basic ELM and regularized ELM tends to be unstable. WRELM was originally proposed for suppressing the influence of outliers in training data [99], where the data samples with high training error are assigned with small weights. In this case, we have

$$\begin{aligned} \mathbf{B}^* &= \arg\min_{\mathbf{B}} \mathcal{C}\|\mathbf{W}\xi\|^2 + \|\mathbf{B}\|^2 \\ &\text{s.t.}\, \mathbf{Y} - \mathbf{HB} = \xi, \end{aligned} \tag{2.12}$$

where $\mathbf{W} = diag\{w_1, \cdots, w_N\}$ with w_i's being the weights imposed on the training errors. From [99], the optimal solution to (2.12) can be derived as

$$\mathbf{B}^* = (\mathbf{H}^T\mathbf{W}^2\mathbf{H} + \frac{\mathbf{I}}{\mathcal{C}})^{-1}\mathbf{H}^T\mathbf{W}^2\mathbf{Y}, \qquad N > L. \tag{2.13}$$

Notably, the weights in Eqn. (2.13) play an important role in the WRELM method. In order to obtain a robust estimation of $\mathbf{W}$, many methods like Huber weights, bisquare weights, Cauchy weights, etc., have been proposed in [99–101], one of which in [99] is

$$w_i = \begin{cases} 1 & |\xi_i/\hat{s}| \leq a \\ \frac{b - |\xi_i/\hat{s}|}{b-a} & a < |\xi_i/\hat{s}| < b \\ 10^{-4} & otherwise \end{cases} \tag{2.14}$$

where $\hat{s}$ is the standard deviation of error variables ξ_i's of the regularized ELM without weights and its robust estimate is $MAD(\xi_i)/0.6745$ where MAD denotes the median absolute deviation. a and b are determined as empirical values of 2.5 and 3, respectively. This choice is reasonable since there are very few residuals larger than $2.5\hat{s}$ for a Gaussian distribution. WRELM can be summarized in Algorithm 1.

Algorithm 1 Procedure for WRELM algorithm.

1: Randomly generate hidden node parameters $\{\mathbf{a}_i, b_i, i = 1, \cdots, L\}$, calculate the hidden layer output matrix $\mathbf{H}$ from Eqn. (2.3);
2: Calculate the initial output weight vector $\mathbf{B}^0$ according to RELM solution in Eqn. (2.9);
3: Calculate the training error variables $\boldsymbol{\xi} = \mathbf{Y} - \mathbf{HB}^0$;
4: Calculate the weight matrix $\mathbf{W}$ with respect to training error variables of step 3, e. g., from Eqn. (2.14);
5: Finally, Eqn. (2.13) will be used for obtaining the WRELM solution.

2.1.4 Outliers Robust Extreme Learning Machine

Notably, since the weights calculation is dependent on the training errors, they have to be predetermined by using previous RELM. Hence, both ELMs (i. e., RELM and WRELM) are included in the procedure. For this reason, the WRELM algorithm has certain drawbacks, summarized as follows,

- Firstly, the computational training time increases inevitably and can be much greater than the basic ELM and the RELM without weights.
- Secondly, error propagation may occur due to the dependence of weights on previous RELM solution.
- Finally, it is possible that some good training samples that are not outliers but have large RELM training errors may be weighted by small values so as to weaken the training.

In order to address these problems, an alternate approach robust to outliers was proposed in [101], the optimization problem in Eqn. (2.8) can be changed as follows

$$\begin{aligned} \mathbf{B}^* \; = \; & \arg\min_{\mathbf{B}} \mathcal{C}\|\xi\|_1 + \|\mathbf{B}\|^2 \\ & \text{s.t.}\, \mathbf{Y} - \mathbf{HB} = \xi, \end{aligned} \tag{2.15}$$

where $\|\cdot\|_1$ stands for the l_1-norm used to enhance the robustness. It can be solved by the augmented Lagrange multiplier (ALM) method. A comprehensive survey on the robust loss function and regularization term was given in [104].

Subsequently, for solving regression problems by ELM methods, substituting the optimal solution weights $\mathbf{B}^*$ of Eqn. (2.15) into Eqn. (2.1) will render the optimal estimation as

$$\hat{\mathbf{y}}_j = \mathbf{h}(\mathbf{x}_j)\mathbf{B}^*, j = 1, \cdots, N. \tag{2.16}$$

As for the binary classification and multi-classification problems, the classifier output can be respectively expressed as

$$\hat{\mathbf{y}}_j = \mathtt{sign}\left(\mathbf{h}(\mathbf{x}_j)\mathbf{B}^*\right), j = 1, \cdots, N, \tag{2.17}$$

and

$$\hat{\mathbf{y}}_j = \mathtt{maxpool}\left(\mathbf{h}(\mathbf{x}_j)\mathbf{B}^*\right), j = 1, \cdots, N, \tag{2.18}$$

where `sign` and `maxpool` denote the symbol function and the max pooling operation, respectively.

2.2 Bidirectional Long-Short Term Memory Networks

LSTM is a type of recurrent neural networks (RNNs), which has been widely studied and implemented in image processing, sentiment analysis, language translation [105] and handwriting recognition, etc. It is a promising technique in sequential data prediction and pattern recognition due to its ability of memorizing the previous states information.

Unlike the traditional LSTM which only has the forward layer, BiLSTM employs both forward and backward layers. The structure of BiLSTM is shown in Fig. 2.2. In BiLSTM [106], the current output y_t is dependent on both the past and the future status. For example, the current output y_t is a function of both the forward layer output $\vec{h}_t$ and the backward layer output $\overleftarrow{h}_t$. The forward layer output $\vec{h}_t$ is dependent on the current input x_t, the past forward layer output $\vec{h}_{t-1}$ and its cell state $\vec{C}_{t-1}$, hence the past status is taken into account. Similarly, the backward layer output $\overleftarrow{h}_t$ is dependent on the current input x_t, the future backward layer output $\overleftarrow{h}_{t+1}$ and its cell state $\overleftarrow{C}_{t+1}$, hence the future status is also considered. Thus, by employing this structure, the ambient information can be exploited and utilized to decide the current output.

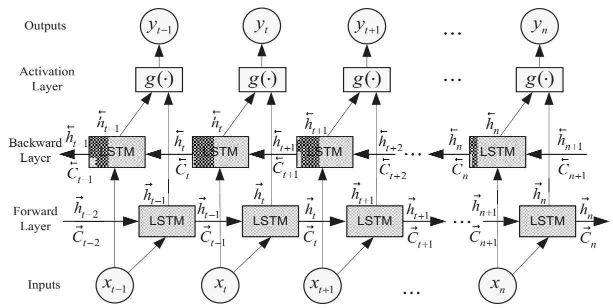

Figure 2.2: Bidirectional LSTM structure.

The core block in BiLSTM is the memory cell as shown in Fig. 2.3, the brief introduction of one memory cell is given as follows.

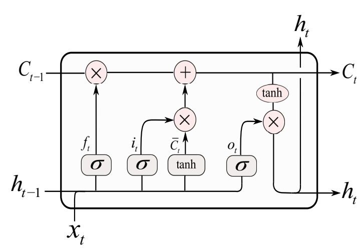

Figure 2.3: Memory cell of LSTM.

A memory cell contains three control gates, namely the input, forget and output gate. They are described as follows:

1. Input gate: It controls whether the memory cell is updated.

$$i_t = \sigma(W^i(h_{t-1}, x_t) + b^i) \tag{2.19}$$

2. Forget gate: It controls if the memory cell is reset.

$$f_t = \sigma(W^f(h_{t-1}, x_t) + b^f) \tag{2.20}$$

3. Output gate: It controls if the current cell state C_t is made visible.

$$o_t = \sigma(W^o(h_{t-1}, x_t) + b^o) \tag{2.21}$$

Besides, the cell state modification $\bar{C}$ is described as:

$$\bar{C}_t = tanh(W^c(h_{t-1}, x_t) + b^c) \tag{2.22}$$

It can be seen that the current gating effects i_t, f_t, o_t and the cell state modification $\bar{C}_t$ are all functions of the previous hidden state h_{t-1} and the current input x_t. Then the current cell state C_t and hidden state vector h_t are expressed as:

$$C_t = f_t C_{t-1} + i_t \bar{C}_t \tag{2.23}$$

$$h_t = tanh(C_t) \times o_t \tag{2.24}$$

The parameters in BiLSTM that can be obtained through training process are $[W^i, b^i]$, $[W^f, b^f]$, $[W^o, b^o]$ and $[W^c, b^c]$. They denote the weights and biases for input gate, forget gate, output gate and the cell state modification respectively. The training process is usually performed by the back-propagation through time (BPTT) algorithm [107].

Chapter 3

Transmission Reduction by Conditional Training based GM and GM-OPELM Data Fusion Schemes*

3.1 Introduction

WSNs are usually formed by a large number of distributed sensor nodes in the cluster formation and each node is equipped with a sensor to measure physical quantities such as light, heat, pressure, etc. The measurements are transmitted to the corresponding sink nodes by the distributed children sensor nodes. As a key infrastructure of IoTs, WSNs have attracted great research interests in multiple fields [108]. Distributed sensors are usually deployed in harsh or isolated environment which is not easily accessible for maintenance. The energy efficiency is a crucial factor in determining the entire lifetime of WSNs. To lower the energy consumption thus extend the lifetime of WSNs, many methods have been proposed, such as the node sleep/active scheme [109], clustering protocol design [110] and data fusion, etc. In WSNs, the power consumed on wireless radio transmissions among nodes dominates the total energy consumption, thus, reducing the transmission workload will greatly

*A version of this chapter has been published as "Conditional training based GM and GM-OPELM data fusion schemes in wireless sensor networks," in *2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM).*

improve the lifetime of the nodes.

Data collected by sensor nodes during continuous sensing periods usually is of high temporal coherence, representing high redundancy in the continuous data sequence. Thus, to transmit all data from the sensor nodes to the sink nodes, it leads to redundancy in data transmissions and high energy consumption. The dual prediction based data fusion schemes provide effective means for reducing the transmissions between data collecting sensors and the corresponding sink node. By running the same data prediction algorithms in both the sensor node and the sink node, data does not need to be transmitted if the prediction error is within the tolerance range. In this case, the predicted data will be taken as the sensed data for data acquisition. Grey prediction model (GM) [111] is effective in predicting time series by using small initial data sequence with low computational complexity. In [112], a GM combing Kalman Filter prediction is proposed which integrates the merit of GM in quick modeling and the advantage of Kalman Filter in processing noise. Recently, machine learning based methods have been widely adopted with improved prediction accuracy. In [113] a method based on Kalman Filter combining support vector regression (SVR) is applied to improve the prediction accuracy and the work in [114] utilizes the deep learning convolutional neural network method to perform data fusion in action recognition. In [115], the GM combined with kernel recursive least-squares (KRLS) fusion method is proposed where the KRLS can adaptively adjust its model coefficient with every input to maintain the prediction accuracy. Furthermore, the method of combining GM and optimally-pruned extreme learning machine (GM-OPELM) [116] has been proposed where the OPELM is trained to correct the prediction error of GM. Experiments show that the GM-OPELM can significantly reduce redundant transmissions and extend the WSNs lifetime.

However, in the scheme in [116], the model training needs to be conducted at the sink node and the model coefficients need to be sent to the sensor node before every prediction. This introduces excessive training time cost and unnecessary energy consumption for training and signal transmissions. In view of this limitation, we propose the conditional training (CT) idea for the prediction based data fusion schemes to reduce the amount of model training and energy consumption for the transmissions. The CT idea can be applied to both the GM scheme and GM-OPELM scheme, and we refer the corresponding new schemes as CT-GM and CT-GM-OPELM, respectively. The main contributions of this chapter can be summarized as follows.

- The proposed CT based schemes only update the model when the prediction error is beyond the user-defined threshold $\boldsymbol{\epsilon}$ instead of training before every prediction. The time and energy cost of sink node on model training and the updated model parameters transmission are dramatically reduced.

- The proposed CT based schemes can improve the rate of acceptable prediction R with respect to different $\boldsymbol{\epsilon}$. R is defined as the ratio of the number of predictions with the error less than $\boldsymbol{\epsilon}$ to the number of all predictions. The improvement in the acceptable prediction rate further leads to the reduction of data transmissions required from the sensor node, thus saving in the energy consumption of sensor node.

- Compared with the GM-OPELM and GM methods, the proposed CT based methods own the advantage of higher rate of acceptable prediction with respect to the different lengths of training set.

This chapter is organized as follows. The GM and OPELM schemes are briefly introduced in Section 3.2. Section 3.3 presents the proposed CT-GM and CT-GM-OPELM schemes. Simulation and performance evaluations are included in Section 3.4 and Section 3.5 concludes the chapter.

3.2 Related Work

3.2.1 Grey Prediction Model

A grey prediction model, denoted as GM(1,1), represents the first order one variable grey prediction model which is widely used in time series prediction due to its less computational burden. The differential equation of GM(1,1) has time-varying coefficients and it will be renewed when new data become available to the prediction model [117]. Following is the brief introduction of GM(1,1) model.

Assume we have a positive time series with n data as:

$$X^{(0)} = [x^{(0)}(1), x^{(0)}(2), \cdots, x^{(0)}(n)] \tag{3.1}$$

We want to predict the value at any time instance $\hat{x}^{(0)}(k)$. Create the accumulating generation operation (AGO) sequence $X^{(1)}$:

$$X^{(1)} = [x^{(1)}(1), x^{(1)}(2), \cdots, x^{(1)}(n)] \tag{3.2}$$

where

$$x^{(1)}(k) = \sum_{i=1}^{k} x^{(0)}(i), \qquad k = 1, 2, \cdots, n. \tag{3.3}$$

Generate a mean value sequence $Z^{(1)}$ from $X^{(1)}$ as:

$$Z^{(1)} = [z^{(1)}(1), z^{(1)}(2), \cdots, z^{(1)}(n)] \tag{3.4}$$

where $z^{(1)}(k)$ is the mean value of two adjacent data,

$$z^{(1)}(k) = 0.5x^{(1)}(k) + 0.5x^{(1)}(k-1) \tag{3.5}$$

Establish the first order differential equation as:

$$\frac{\partial x^{(1)}(k)}{\partial k} + ax^{(1)}(k) = b \tag{3.6}$$

Substitute the $x^{(1)}(k)$ with the mean value $z^{(1)}(k)$, we have

$$x^{(0)}(k) + az^{(1)}(k) = b \tag{3.7}$$

Apply ordinary least squares method to find the estimated value of a and b.

$$[\hat{a}, \hat{b}]^T = (B^T B) B^T Y \tag{3.8}$$

where

$$B = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \vdots & \vdots \\ -z^{(1)}(n) & 1 \end{bmatrix}, \qquad Y = \begin{bmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \vdots \\ x^{(0)}(n) \end{bmatrix} \tag{3.9}$$

Substitute $\hat{a}$ and $\hat{b}$ into the solution of (3.6), we have the predicted value of $\hat{x}^{(1)}(k)$ as

$$\hat{x}^{(1)}(k) = \left(x^{(0)}(1) - \frac{\hat{b}}{\hat{a}} \right) e^{-\hat{a}(k-1)} + \frac{\hat{b}}{\hat{a}} \tag{3.10}$$

Then by applying inverse AGO method, the prediction of $x^{(0)}$ at the k-th time instance can be written as:

$$\hat{x}^{(0)}(k) = \left(x^{(0)}(1) - \frac{\hat{b}}{\hat{a}} \right) e^{-\hat{a}(k-1)}(1 - e^{\hat{a}}) \tag{3.11}$$

3.2.2 Optimally-Pruned Extreme Learning Machine

OPELM is a modified method based on ELM with additional steps to make it more robust and generic. ELM algorithm may not be very accurate when the input data set contains irrelevant elements to the output or the observations are to some extent correlated. To overcome this kind of drawbacks, OPELM is proposed [118] where a pruning procedure is applied to eliminate the irrelevant variables via pruning of the related hidden layer neurons. In this Chapter, the RELM with optimally-pruning process is employed and the main steps of OPELM [119] are shown in Algorithm 2.

3.3 The Proposed CT-GM and CT-GM-OPELM Method

In this section the proposed conditional training based algorithms are presented. Although we focus on the CT-GM-OPELM, the CT-GM scheme is

Algorithm 2 Optimal-Pruning process of OPELM.

1: Set the RELM structure as 1 input nodes, 1 output node and the maximum permitted hidden nodes number N;
2: Train RELM network (Algorithm 1) with 1 hidden node
3: Add one more hidden node together with h_1 to train RELM network (Algorithm 1) with 2 hidden nodes. The newly added node is selected from the hidden nodes other than h_1. Evaluate the usefulness of all the networks with 2 hidden nodes and choose the best one. The corresponding selected hidden nodes are denoted as $[h_1, h_2]$;
4: Repeat with k hidden nodes, $k = [3, 4, \cdots, N]$, until the maximum hidden nodes number N is reached.
5: Perform leave-one-out (LOO) validation on the N trained RELM networks, the one with least LOO error is chosen and the corresponding L hidden nodes are selected.
6: The hidden nodes other than the L chosen ones are pruned and the best RELM structure is achieved.

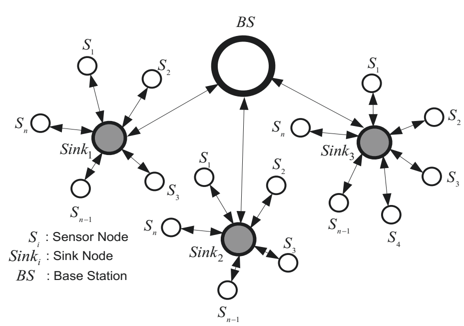

Figure 3.1: Structure of WSNs.

a stand-alone part, which can be integrated with other models than ELM. CT-GM-OPELM data fusion method can be applied in the centralized WSNs environment as shown in Fig. 3.1. Each sensor node communicates with its own sink node to transmit sensed data and receive the overhead information such as renewal of model parameters. The sink nodes communicate with the base station to access the internet. The proposed CT-GM-OPELM scheme is a dual prediction method which requires prediction models in the sensor node and the sink node are highly synchronized. Once the prediction error is beyond the user-defined error threshold, a transmission will take place for the sensed data to be sent from the sensor node to its sink node, then the sink node will concatenate the received sensed data instead of the predicted one into the sequence of the fused data. Considering the limitation of computational capability of the sensor node, the model training process is preformed at the sink node which is usually more powerful in terms of computational capability, data buffer size and energy storage. When receiving the newly sensed data from a sensor, a new round of model training will be implemented by the sink node and then the updated model parameters are sent back to the sensor to keep the models at both sides the same.

By taking a pair of sink node $Sink_1$ and sensor node S_1 as an example, the main steps of the proposed CT-GM-OPELM scheme can be described as follows:

1. Initialization: $Sink_1$, which is a sink node, broadcasts the *start* instruction and the length of training set L to the sensor S_1. S_1 starts to collect L initial sensed data points $[x_1, \cdots, x_L]$ and transmit the data sequence to $Sink_1$. Set the initial state of training set $\boldsymbol{X}_T$ and fused data sequence $\boldsymbol{X}^f$ as $\boldsymbol{X}_T = \boldsymbol{X}^f = [x_1, \cdots, x_L]$.

2. Training the initial GM-OPELM model: Take $\mathbf{X}_T$ as the primitive data sequence to form the GM model. Find the GM parameters $\hat{a}$ and $\hat{b}$ through the ordinary LS estimation of the established first order difference equation. The predictions produced by GM denoted as $\tilde{\mathbf{X}}_{gm} = [\tilde{x}_1, \cdots, \tilde{x}_L]$ can thus be obtained by Eqn.(3.11). Take the GM outputs $\tilde{\mathbf{X}}_{gm}$ as the input of the ELM network, and $\mathbf{X}_T$ as the target values to train the ELM networks.

3. Parameter transmission: After the initial GM-OPELM model is trained in $Sink_1$, the corresponding model parameters are transmitted to S_1. Also transmitted from $Sink_1$ is a user-defined $\boldsymbol{\epsilon}$ which is the error threshold for determining if the error between the predicted value and the sensed data is acceptable.

4. Prediction: The data prediction is conducted at both S_1 and $Sink_1$

synchronically. The prediction of the i-th ($i \geq L+1$) data value is based on the past L sequenced data in $\mathbf{X}^f$, i.e., $x^f_{i-1}, \cdots, x^f_{i-L}$, and the trained model. At the S_1 side, a new sensed data x_i is collected at time i. Assume that S_1 has the noise filtering process and x_i can be referred as the true value of sensed data.

5. Conditional model retraining: The prediction error can be calculated by $error_i = |x_i - \hat{x}_i|$ at the sensor node. If $error_i \leq \boldsymbol{\epsilon}$, the predicted value $\hat{x}_i$ is acceptable, thus no transmission will take place. The new predicted data will be concatenated into the fused data sequence as $\mathbf{X}^f = [x^f_1, \cdots, x^f_{i-1}, \hat{x}_i]$ at $Sink_1$ and the prediction continues at both sides. On the other hand, it appears that $error_i > \boldsymbol{\epsilon}$, the prediction $\hat{x}_i$ will not be accepted. In this case, a transmission will take place to transmit the sensed data x_i to $Sink_1$. The $Sink_1$ will concatenate x_i into the fused data sequence as $\mathbf{X}^f = [x^f_1, \cdots, x^f_{i-1}, x_i]$.

   Further, since the prediction error is beyond $\boldsymbol{\epsilon}$, the prediction model is to be retrained. A new training data set is formed by the L lately sequenced data in the fused data sequence $\mathbf{X}^f$ as $\boldsymbol{X}_T = [x^f_{i-L+1}, \cdots, x^f_i]$ to obtain a new GM and train the OPELM following the same training procedure explained in Step 2. After the new model is trained, the new model parameters are transmitted from $Sink_1$ to S_1. Then at both ends, Steps 4 and 5 are repeated for the prediction and conditional training of Time $i+1$.

The final result of fused data sequence $\mathbf{X}^f$ contains both real sensed data x_i and predicted data $\hat{x}_i$. The less sensed data it contains, the less transmissions occur, thus more energy will be saved. The performance in terms of the rate of acceptable prediction and training times are evaluated in Section 3.4.

The overall procedures of the proposed method CT-GM-OPELM is summarized in Algorithm 3. The CT-GM algorithm can be referenced in Algorithm 3 by ignoring Step 3.

Algorithm 3 CT-GM-OPELM data fusion method.

1: S_1 starts to collect L initial data points $[x_1, \cdots, x_L]$ and send to $Sink_1$, set the initial state of training set $\boldsymbol{X}_T$ and fused sequence $\boldsymbol{X}^f$ as $\boldsymbol{X}_T = \boldsymbol{X}^f = [x_1, \cdots, x_L]$;
2: Get $\hat{a}$ and $\hat{b}$ of GM from $\boldsymbol{X}_T$, generate GM predictions $\tilde{\mathbf{X}}_{gm}$ from Eqn. (3.11);
3: Train ELM with input $\tilde{\mathbf{X}}_{gm}$ and target $\boldsymbol{X}_T$;
4: **for** $i = L+1 : N$ **do**
5: $\quad$ $Sink_1$ and S_1 predict the next value $\hat{x}_i$
6: $\quad$ S_1 collects data x_i
7: $\quad$ **if** $|x_i - \hat{x}_i| \leq \boldsymbol{\epsilon}$ **then** $\boldsymbol{X}^f = [x_1^f, \cdots, x_{i-1}^f, \hat{x}_i]$
8: $\quad$ **else** $\boldsymbol{X}^f = [x_1^f, \cdots, x_{i-1}^f, x_i]$
9: $\quad\quad$ $\boldsymbol{X}_T = [x_{i-L+1}^f, \cdots, x_i^f]$, do Step 2 and Step 3;
10: $\quad$ **end if**
11: **end for**

3.4 Simulation

3.4.1 Simulation Environment

To test the performance of the proposed CT-GM-OPELM and CT-GM schemes in terms of the rate of acceptable prediction and time efficiency in WSNs environment, experiments are conducted on a real data set available on the website of `Intel Berkeley Research Lab`[†]. The data set contains the temperature, humidity and light data collected by 54 distributed sensors in the Berkeley research lab. The following experiment results are based on the temperature and humidity data collected by the third sensor node. Comparisons are made between CT-based schemes, i.e., the CT-GM-OPELM and CT-EM methods, and non-CT based, i.e., the GM-OPELM, GM methods. A total of 3000 samples for each feature are selected for the experiment. The investigated error threshold is set to be $\boldsymbol{\epsilon} \in [0.05, 0.5]$ with the step size 0.05. The training sequence length is set to be 50 so that there are 2950 data points to be predicted under the condition of each error threshold. Simulations are conducted on a computer with a 3.4GHz Intel(R) Core(TM) i5-8400 CPU and 8GB RAM with Matlab R2018a (64bit).

†http://db.csail.mit.edu/labdata/labdata.html

3.4.2 Rate of Acceptable Prediction

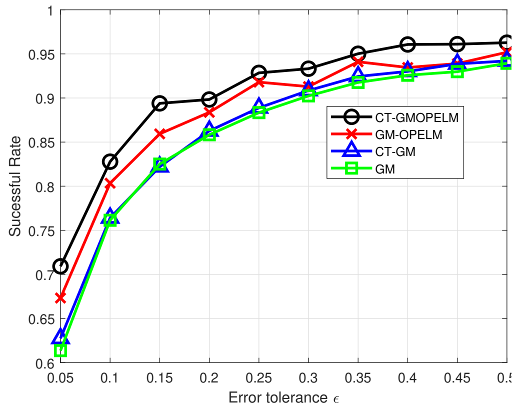

Figure 3.2: Successful Rate of temp.

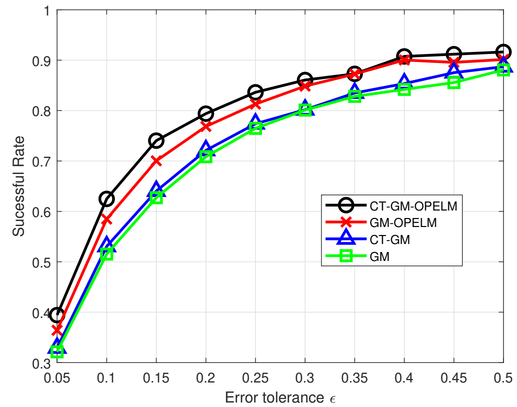

Figure 3.3: Successful Rate of humi.

When the prediction error is less than $\boldsymbol{\epsilon}$, a prediction is regarded as acceptable prediction. Define the ratio between the number of acceptable prediction and total number of predictions as the rate of acceptable prediction R. Thus, if N is the total number of samples needed to be sensed, the counts of transmissions C occurs at S_1 can be calculated by $C = N * (1 - R)$, where R is the acceptable prediction rate. Fig. 3.2 and Fig. 3.3 demonstrate that the proposed CT based methods outperform the corresponding non-CT based methods. The CT-GM-OPELM method achieves the best acceptable prediction rate among all the algorithms. Compared with the GM-OPELM method, in Fig. 3.2, when $\boldsymbol{\epsilon} = 0.15$, the acceptable prediction rate rises up from 0.86 to 0.89. In this case, where $N = 2950$, the number of transmissions occurs at S_1 is reduced by 88% if CT is applied. The CT-GM is slightly better than GM in terms of the rate of acceptable prediction.

Furthermore, it can be observed that the methods combining ELM can achieve higher acceptable prediction rate than the conventional GM only methods for most of the $\boldsymbol{\epsilon}$ values, which indicates that utilizing ELM can increase the prediction accuracy of GM model.

3.4.3 Number of Training

Fig. 3.4 and Fig. 3.5 demonstrate the number of model training in the process of predicting the 2950 samples. For the GM and GM-OPELM methods, the number of training are both 2950 due to the model updating before each prediction. It can be seen that the proposed CT based methods can dramatically decrease the number of model update. Thus, the energy consumed on training and updated model transmission is dropped. For example, when $\epsilon = 0.2$, in Fig. 3.4 only 313 times of training will take place by CT-GM-OPELM. In this case the energy consumed by $Sink_1$ for training will dropped to only 10.6% of that under the GM-OPELM method. It can be observed that the training number of CT-GM also significantly drops but not as much as the CT-GM-OPELM due to the better prediction accuracy performance owned by ELM based methods.

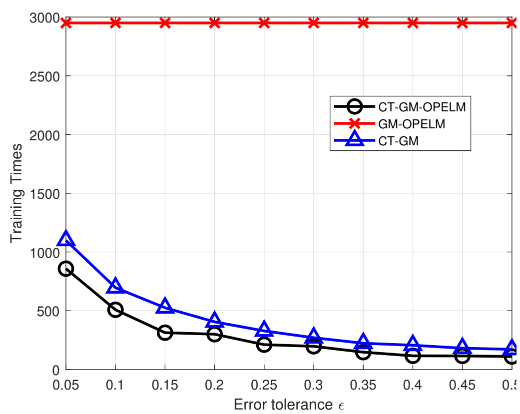

Figure 3.4: Training number of temp.

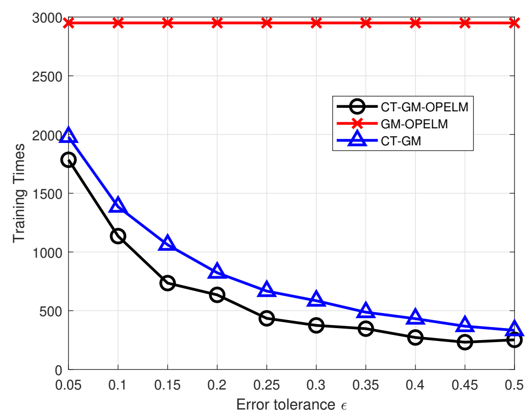

Figure 3.5: Training number of humi.

3.4.4 Prediction Time Consumption

Fig. 3.6 and Fig. 3.7 demonstrate the time consumption plots of predicting 2950 data samples with respect to different error threshold ϵ varying from 0.05 to 0.5.

Further, in Table 3.1, the time consumption of different algorithms when $\epsilon = 0.2$ is shown. It can be seen that compared with non-CT methods, the proposed ones have the advantage of time efficiency. Compared with the GM-OPELM method, the CT-GM-OPELM method can save almost 80% of the running time. The CT-GM takes about half of the running time comparing with GM. Furthermore, the non-ELM based methods are much faster than the ELM combined algorithms. The fast running speed and fairly good acceptable prediction rate make CT-GM method a competitive option when the balance between data acquisition speed and energy consumption is to be considered.

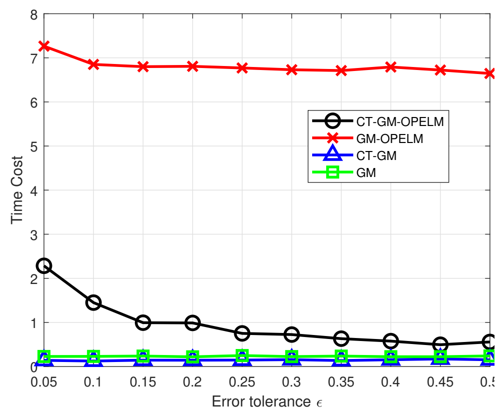

Figure 3.6: Time cost of temp.

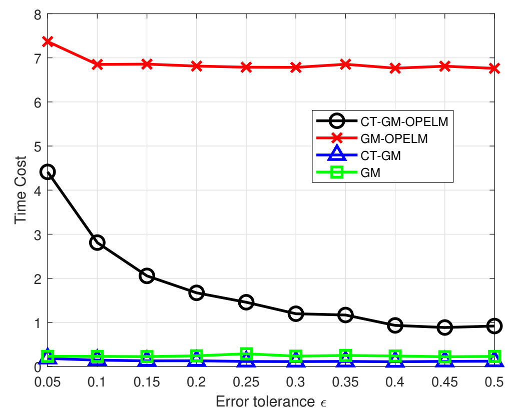

Figure 3.7: Time cost of humi.

Table 3.1: Prediction time comparisons among different algorithms.

Data Type	*GM-OPELM*	*CT-GM-OPELM*	*GM*	*CT-GM*
Temperature	6.81s	0.99s	0.22s	0.14s
Humidity	6.81s	1.67s	0.24s	0.13s

3.4.5 Effects of Training Set Length

Fig. 3.8 and Fig. 3.9 are the plots of the rate of acceptable prediction with respect to different training lengths when the error threshold is set to be $\boldsymbol{\epsilon} = 0.2$. The figures demonstrate that the proposed CT based methods can achieve obviously better acceptable prediction rate than the non-CT methods. Among all the methods, the proposed CT-GM-OPELM has the best acceptable rate with respect to most of the training length conditions. The performance gap between ELM based and the corresponding non-ELM based methods enlarges as the training set length increases. However, in Fig. 3.9, when the training length is less than 16, the rate of acceptable prediction performance of non-ELM based methods is better than the corresponding ELM based methods. It shows that the ELM based methods are not always outperforming the non-ELM based methods in terms of the acceptable rate under the condition of different training lengths and this phenomenon is data set dependable. Furthermore, the proposed CT-GM-OPELM is more steady than other methods and has little performance drops. It indicates that CT-GM-OPELM owns good robustness of acceptable rate to different training set length.

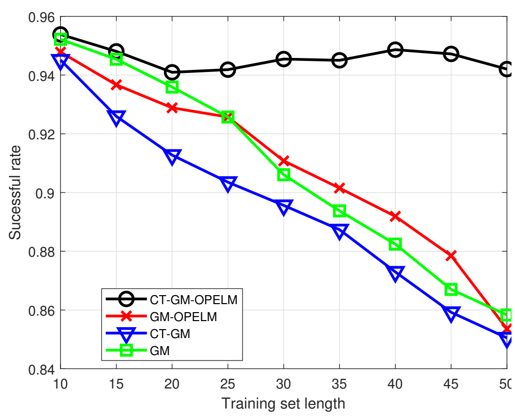

Figure 3.8: Training robust. of temp.

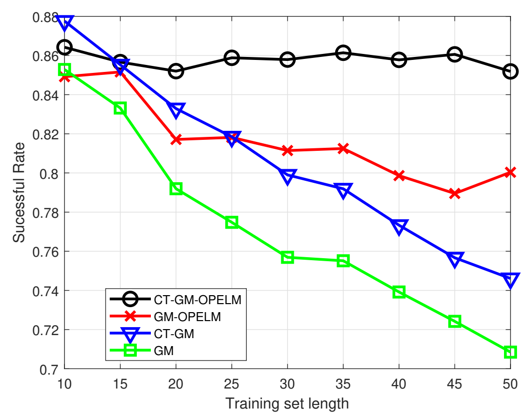

Figure 3.9: Training robust. of humi.

3.5 Conclusion

This chapter proposes the conditional training based data fusion methods, i.e., CT-GM and CT-GM-OPELM. Compared with the non-CT based methods, the advantages of the proposed ones can be summarized as follows. Firstly, the proposed methods can decrease the sensor node's energy cost on data transmission by improving the rate of acceptable prediction. Secondly, the proposed methods can dramatically decrease the unnecessary model training and broadcasting energy cost of the sink node by introducing conditional training scheme. Thirdly, the proposed methods greatly improve the time efficiency via reducing the number of time consuming training processes. Finally, the proposed methods own higher rate of acceptable prediction with respect to different lengths of training set than existing non-CT based methods. Notably, the proposed CT-GM-OPELM method has good robustness in the acceptable prediction rate to different training set lengths.

Chapter 4

Communication Channel Equalization and Signal Detection by ELM-based Regressors Methods*

4.1 Introduction

Orthogonal frequency division multiplexing (OFDM) is a popular digital signal modulation method where channel equalization is usually conducted before symbol demodulation and detection to compensate the multi-path interference at the reception. To perform channel equalization in OFDM systems, several ML based equalization schemes have been developed [123–125]. In [123], a complex radial basis function neural network (NN) was used as a regression based channel equalizer which performs Bayesian estimation. SVM as one of the common machine learning techniques was initially adopted for nonlinear detection in [124] and successfully applied to nonlinear equalization problems in coherent optical OFDM systems [125].

Although NN and SVM have been widely applied, there still exist some challenging issues such as slow learning speed and inevitable human intervention. The merits of very fast learning speed and little human intervention owned by ELM [98] make it a desirable choice in equalization applications. In

∗A version of this chapter has been published as "Channel equalization and detection with ELM-based regressors for OFDM systems," in *IEEE Communications Letters*

[126], fully complex ELM (CELM) was used to construct a regression based nonlinear channel equalizer in the time domain. However, the performance of that method deteriorates greatly when extended to OFDM systems. This is because the time domain ELM cannot effectively capture the frequency selective characteristics of the individual narrow-band subchannel, especially when the frequency selectivity is strong. The method in [127] jointly addresses the equalization and symbol detection for OFDM systems as a multi-class classification task by applying split-complex ELM (SCELM) in the frequency domain. However, in this design, the function of symbol slicer is embedded into the networks which may cause more computational burden and errors. Furthermore, in [128] the single CELM network was proposed for OFDM systems. The symbol error rate (SER) performance of this design degrades obviously as the subchannel number increases. Moreover, the required numbers of hidden neurons and training data are large, leading to high computational complexity and slow learning speed.

In view of these limitations, we propose the parallel structured multiple SCELM (Multi-SCELM) channel equalization and symbol detection method for OFDM systems. The main contributions compared to the existing work are as follows.

- The proposed Multi-SCELM method outperforms the time domain CELM method in [126], the frequency domian SCELM classifier based method in [127] and single CELM method in [128] in terms of computational complexity, detection accuracy, activation function adaptability, training set length and robustness to subchannel numbers.

- The proposed Multi-SCELM method is extended to multiple CELM (Multi-CELM) with slight complexity increase but better detection performance.

The rest of this chapter is organized as follows. In Section 4.2, the proposed method is presented, followed by complexity comparisons. Simulation results are presented in Section 4.3, and Section 4.4 concludes the chapter.

4.2 ELM Based Equalization and Detection

In OFDM systems, as shown in Fig. 4.1, binary data bits are firstly mapped to M-ary modulated symbols over N narrowband channels in frequency domain where the modulation order is denoted as M. Denote the K transmitted symbols assigned to the nth subchannel as $\mathbf{X}^n = \{X_1^n, \cdots, X_k^n, \cdots, X_K^n\}$, where X_k^n is the kth symbol on the nth subchannel. After the N-point fast inverse Fourier transform (IFFT) and insertion of cyclic prefix (CP), the kth time domain transmitted signal stream $\mathbf{x}_k$ is produced and totally K time domain streams are transmitted. At the receiver side, the kth time domain signal stream $\mathbf{y}_k$ is received, which is corrupted by additive white Gaussian noise (AWGN). After CP removal and FFT, the frequency domain symbol stream is obtained. Denote the received symbols on the nth subchannel as $\mathbf{Y}^n = \{Y_1^n, \cdots, Y_k^n, \cdots, Y_K^n\}$. To mitigate inter-symbol interference, Y_k^n is equalized to get the estimation of transmitted symbols $\tilde{Y}_k^n$. Then $\tilde{Y}_k^n$ is fed to the symbol slicer, which produces the hard detection of the transmitted symbols, denoted as $\hat{X}_k^n$. The transmitted binary data bits can be decoded from $\hat{X}_k^n$ accordingly.

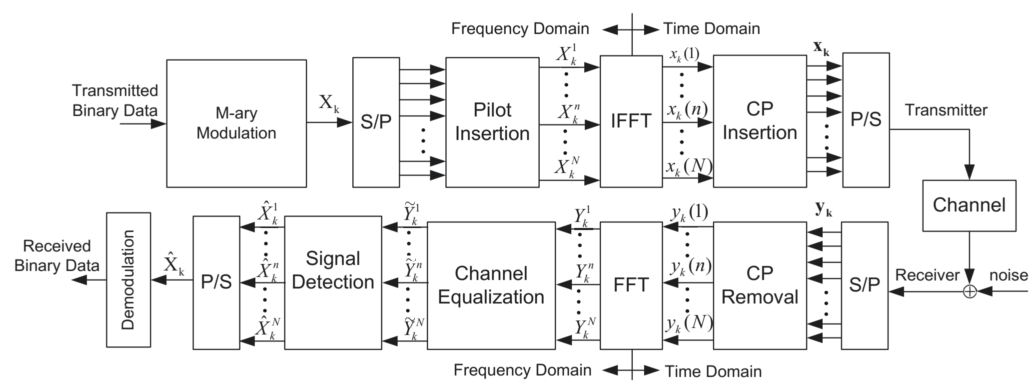

Figure 4.1: Diagram of OFDM system.

4.2.1 The Proposed Multiple SCELM Method

It is known that ELM operates as generalized single hidden layer feedforward NN with no need to tune the parameters of the hidden layer. As proposed in [97], the parameters of hidden nodes (input weights and biases) can be randomly assigned following any continuous distribution function and remain fixed for calculating the output layer weights afterwards. The output weights are analytically determined by least-squares solutions of a general system of linear equations. Under mild smoothness condition on the system, the asymptotic convergence of the ELM is guaranteed [97]. Generally speaking, the ELM operates as an universal approximator, which can produce superior approximation performance in most cases and can learn thousands of times faster than other conventional algorithms, making it a desirable choice for equalization.

Inspired by the structure shown in [123], we propose a Multi-SCELM equalization combining the minimum-distance detection method for OFDM systems. The proposed structure for one of the N subchannels is shown in Fig. 4.2 and the superscript (n) for the subchannel index is omitted in the following. The four parts of the proposed equalization and detection method are described as follows.

- Input layer: The input layer contains 2 nodes corresponding to the real and imaginary part of the kth received frequency-domain symbol Y_k, the input vector can be written as $\mathbf{Y}_k = [\Re(Y_k), \Im(Y_k)]^T$.

- Hidden layer: The hidden layer contains L hidden nodes with activation function $g(x) : R^2 \to R$. The output of the lth hidden node is given by

$$h_l = g({\mathbf{a}_l}^T \cdot \mathbf{Y}_k + b_l), \tag{4.1}$$

  where $l = 1, \cdots, L$ and $\mathbf{a}_l$ is the 2×1 column vector of the weights connecting the lth hidden node and the two input nodes. b_l is the bias of the lth hidden node. Here $\mathbf{a}_l$ and b_l are randomly assigned following uniform distribution on $[-0.2 \quad 0.2]$ and kept fixed afterwards.

- Output layer: The output layer contains 2 nodes corresponding to the real and imaginary parts of the equalized symbol $\tilde{Y}_k$, which is the estimation of the kth transmitted symbol in the frequency domain. Then, the output of an ELM subnetwork with L hidden nodes can be written by

$$\tilde{\mathbf{Y}}_k(\mathbf{Y}_k) = \sum_{l=1}^{L} h_l \boldsymbol{\beta}_l^T = \mathbf{h}_k \mathbf{B}, \tag{4.2}$$

where $\tilde{\mathbf{Y}}_k = [\Re(\tilde{Y}_k), \Im(\tilde{Y}_k)]$. $\mathbf{h}_k$ is the row vector of the L hidden nodes outputs when the input is the $\mathbf{Y}_k$. $\mathbf{B}$ is the $L \times 2$ output weights matrix connecting hidden layer and output layer. $\mathbf{B}$ can be written as $\mathbf{B} = [\boldsymbol{\beta}_1 \cdots \boldsymbol{\beta}_l \cdots \boldsymbol{\beta}_L]^T$, where $\boldsymbol{\beta}_l = [\beta_{l,1}, \beta_{l,2}]^T$ is the output weights vector connecting the lth hidden node and the 2 output nodes. The output weights matrix $\mathbf{B}$ can be obtained by training the network, which will be explained in a latter part of this section.

- Symbol slicer: Minimum-distance decision based symbol slicer is used to find the symbol in the constellation alphabet that has the minimum Euclidean distance with the equalized symbol $\tilde{Y}_k$. The corresponding symbol is taken as the hard detection result $\hat{X}_k$. That is $\hat{X}_k = \underset{X \in \mathcal{A}}{argmin} \left| \tilde{Y}_k - X \right|$, where $\mathcal{A}$ is the constellation alphabet, e.g., $\{1+j, -1+j, -1-j, 1-j\}$ for 4-QAM modulation.

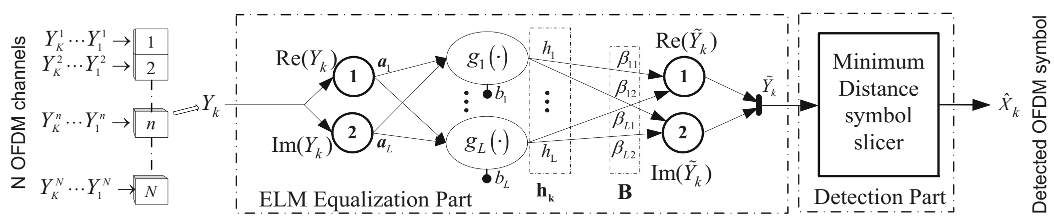

Figure 4.2: The proposed SCELM regressor structure for one subchannel in the frequency-domain.

To train the n-th ELM subnetwork, i.e., the output weights matrix $\mathbf{B}$, pilot signals are used and the training blocks are time-division multiplexed with the data blocks. Assume that there are I pilots and K data symbols assigned to one subchannel, thus the total block length is $J = I + K$. Denote the pilot symbols for training the n-th subnetwork as $\mathbf{X}_p = [\mathbf{X}_{p1} \cdots, \mathbf{X}_{pi}, \cdots, \mathbf{X}_{pI}]^T$

and the corresponding received symbols as $\mathbf{Y}_p = [\mathbf{Y}_{p1} \cdots, \mathbf{Y}_{pi}, \cdots, \mathbf{Y}_{pI}]^T$, where $\mathbf{Y}_{pi}$ is the n-th point of the FFT of the received N point time domain signal after CP removal. Let

$$\mathbf{H}_p = \begin{bmatrix} \mathbf{h}_{p1} \\ \vdots \\ \mathbf{h}_{pi} \\ \vdots \\ \mathbf{h}_{pI} \end{bmatrix} = \begin{bmatrix} g(\mathbf{a}_1^T \mathbf{Y}_{p1} + b_1) \cdots g(\mathbf{a}_L^T \mathbf{Y}_{p1} + b_L) \\ \vdots \quad \cdots \quad \vdots \\ g(\mathbf{a}_1^T \mathbf{Y}_{pi} + b_1) \cdots g(\mathbf{a}_L^T \mathbf{Y}_{pi} + b_L) \\ \vdots \quad \cdots \quad \vdots \\ g(\mathbf{a}_1^T \mathbf{Y}_{pI} + b_1) \cdots g(\mathbf{a}_L^T \mathbf{Y}_{pI} + b_L) \end{bmatrix} \tag{4.3}$$

$\mathbf{H}_p$ denotes the $I \times L$ hidden layer outputs matrix in the training phase. The I rows of $\mathbf{H}_p$ are the hidden layer output vectors corresponding to the I inputs in $\mathbf{Y}_p$ during training.

The cost function is established as follows,

$$Minimize : \mathcal{L}_{ELM} = \frac{1}{2}||\mathbf{B}||^2 + \frac{C}{2}||\mathbf{X}_p - \mathbf{H}_p\mathbf{B}||^2, \tag{4.4}$$

which considers the empirical and structural risk of the proposed ELM. Solving the regularized least squares optimization problem in (4.4), we have

$$\mathbf{B} = \left(\mathbf{H}_p^T \mathbf{H}_p + \frac{\mathbf{I}_L}{C}\right)^{-1} \mathbf{H}_p^T \mathbf{X}_p, \tag{4.5}$$

where C is the coefficient for balancing the empirical and structural risks, $\mathbf{I}_L$ is the L dimensional identity matrix. The overall procedures are summarized in Algorithm 4.

Algorithm 4 ELM equalization and detection for OFDM.

1: Set the ELM structure of one subcarrier as 2 input nodes, 2 output nodes and L hidden nodes;
2: Randomly assign the input weights $\mathbf{a}_l$ and hidden nodes biases b_l, and keep them fixed afterwards;
3: Calculate the hidden layer output matrix $\mathbf{H}_p$ based on Eq. (4.3) using training data input;
4: Calculate the output weight $\mathbf{B}$ by Eq. (4.5).
5: Take $\mathbf{Y}_k$ as input to the trained ELM in Eq. (4.2) to find the equalized symbol $\tilde{\mathbf{Y}}_k$ for all K received symbols.
6: Apply minimum-distance based symbol slicer to find the hard detection $\hat{X}_k$ of every $\tilde{Y}_k$.

4.2.2 Extention to Multiple Fully Complex ELM

The proposed multiple SCELM method can be extended to multiple fully complex ELM (Multi-CELM) by combining the two split real valued neurons of the input and output layers into a complex neuron. Accordingly, all the weights and biases are complex values. The algorithm is same as shown in Algorithm 1. The CELM method has higher computational complexity than the SCELM one but with better detection accuracy as shown in Section 4.2.3 and Section 4.3.

4.2.3 Complexity Analysis and Comparisons

For the computation complexity analysis, the number of multiplications is used. For the proposed Multi-SCELM method, in the learning phase, the number of multiplications needed for Steps 3 and 4 are $N(L^3/3+2IL^2+L^2/2+4IL-5L/6)$. For the data transmission phase of Step 5, $4KLN$ multiplication are needed. Thus, the total complexity is as shown to be $N(L^3/3+2IL^2+L^2/2+4JL-5L/6)$. The complexity results for other methods can be similarly obtained and the results are summarized in Table 4.1. To further understand the comparison, in Table 4.1, the ratio of complexity measure of each method to the proposed Multi-SCELM method is also shown with different parameter settings. For $Ratio1$, the parameters are set as in Table 4.2. In addition, the number of hidden neurons in the proposed method and [127] are set to be $L = 40$ and the size of the training data sets is $I = 200$. $Ratio2$ shows the ratio when modulation order is $M = 64$. $Ratio3$ is the ratio when subchannels number is $N = 128$.

The proposed Multi-SCELM has the least multiplication complexity among all the methods. The extended Multi-CELM method has higher complexity due to the complex value operations but it is still lower than other CELM methods in [126] and [128]. Notably, as shown in $Ratio2$, the complexity of the method in [127] may further increase when high modulation order M applies. As shown in $Ratio3$, the complexity of the proposed ones drop pro-

Table 4.1: Multiplications complexity of various methods.

Proposed Multi-SCELM method	Ratio1	Ratio2	Ratio3
$N(\frac{L^3}{3} + 2IL^2 + \frac{L^2}{2} + 4JL - \frac{5L}{6})$	1	1	1
Proposed Multi-CELM method	Ratio1	Ratio2	Ratio3
$N(8IL^2 + 8JL + \frac{11L^3}{3} + L^2 - \frac{5L}{3})$	3.72	3.72	3.72
The time-domain CELM in [126]	Ratio1	Ratio2	Ratio3
$\frac{2NIL^2}{15} + \frac{61NJL}{15} + \frac{11L^3}{3} + L^2 - \frac{5L}{3}$	53.71	53.71	70.48
The SCELM classifier Method in [127]	Ratio1	Ratio2	Ratio3
$N(\frac{L^3}{3} + 2IL^2 + \frac{L^2}{2} + (M+2)JL - \frac{5L}{6})$	1.11	4.48	4.48
The Single CELM Method in [128]	Ratio1	Ratio2	Ratio3
$8NJL + 8IL^2 + \frac{11L^3}{3} + L^2 - \frac{5L}{3}$	35.33	35.33	59.42
The MMSE Method	Ratio1	Ratio2	Ratio3
$5J(\frac{7N^3}{3} + N^2 - \frac{N}{3})$	1076	1076	269.46

portionally to the decrease of N which makes the complexity advantage over methods in [126] and [128] more prominent. Furthermore, comparing with the benchmark MMSE method, the complexity advantage of the proposed methods is most significant.

4.3 Simulations

In this section, the SER simulations are carried out with respect to different SNRs, various activation functions, I/K ratios and multiple number of subchannels. The parameter settings are shown in Table 4.2 and following methods are used for the comparisons.

- CELM method in time domain [126]: it adopts single CELM network with 60 input and 1 output neuron.
- Multiple SCELM classifiers in the frequency domain [127]: it uses N networks with 2 input neurons and M output neurons in accordance to the modulation order.
- CELM method in frequency domain [128]: it adopts single CELM network with N input and N output neurons.

- MMSE equalizer in the frequency domain: this is used as an ideal benchmark, where perfect channel state information (CSI) and SNRs are assumed to be known.

Table 4.2: Simulation parameters.

Parameters	Values	Parameters	Values
Pilot number	I	Data number	$K = 1000$
Channel model/taps	Rayleigh/60	Subchannels	N=256
Runs of simulations	100	Modulation	QPSK
Hidden neurons [126],[128]	1000	Data rate	500Mbits/s

Fig. 4.3 gives the SER performance of different schemes when $I/K = 0.1$ and the activation function is '*sigmoid*'. Since the performance of the method in [128] is highly dependent on the training length, we display three plots under different I/K. The result indicates that the proposed Multi-SCELM and extended Multi-CELM both achieve better SER performance than other ELM based methods. They outperform the method in [127] with the gain about $8dB$ and $13dB$ respectively in different SNRs environments. The method in [128] fails to achieve comparable performance when SNR is above $20dB$ even under much higher I/K values. The advantage of proposed methods results from the parallel structure of equalizing each subchannel individually while the method in [128] uses a single network for all subchannels, thus the equalizer performance may be degraded in frequency selective channel. The time domain CELM method in [126] ceases to be effective in strong frequency selective OFDM systems. The extended Multi-CELM has the closest performance to the perfect CSI MMSE equalizer.

Fig. 4.4 shows the comparison of SER performance against various activation functions when $SNR = 20dB$ and $I = 100$. It shows that when choosing '*Atanh*' , the method in [127] degrades greatly and the method in [126, 128] fails to give satisfied results. Nevertheless, among the ELM based methods, the proposed one and its extension always have better performance and robustness to the selection of activation functions.

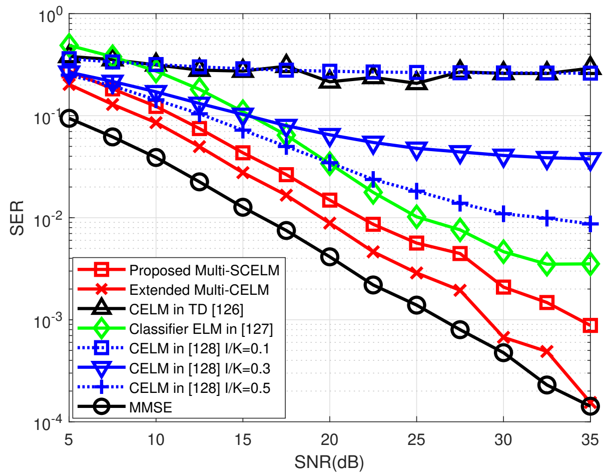

Figure 4.3: SER performance comparisons against SNR.

Fig. 4.5 evaluates the SER performance as the training/testing ratio I/K increases conditioned at $SNR = 20dB$. It can be observed that the multiple subnetworks based methods (a.k.a., the proposed methods and the method in [127]) have better performance than the single network based method in [126] and [128]. They achieve better SER with smaller training set. The proposed methods need even less training data than the method in [127], especially the Multi-CELM method. In other words, by applying the proposed method, higher transmission efficiency and data throughput may be obtained.

To evaluate the generalization performance in OFDM systems, the SER against different subchannel numbers are shown in Fig. 4.6. When $N = 32$ or 64, the number channel taps is set to be 6 or 12; for other values of N, the number of channel taps is set to be 60. It can be observed that the proposed methods and the method in [127] have superior performance due to the multiple ELM structure. The method in [126] has considerably higher SER. The method in [128] has significant performance drop as the number of subchannels increases while the performance of proposed methods drops only slightly.

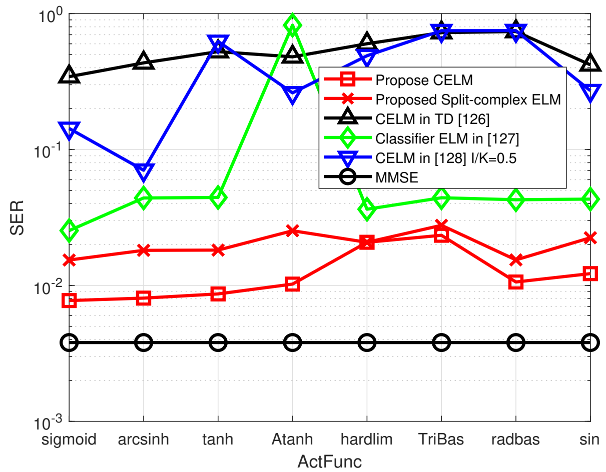

Figure 4.4: SER performance against different activation functions.

4.4 Conclusion

In this chapter, a multiple ELM regressors based channel equalization combined detection structure is proposed for OFDM systems in real and complex domains over strong frequency selective channels. Firstly, the the proposed methods greatly improve the detection performance compared with existing ELM based equalization methods. Multiple simulation environments have been considered in terms of various of SNR values, activation functions, the ratios of training/testing and subchannel numbers. Secondly, the complexity of the proposed methods is lower than the existing methods and the advantage further outstands as the modulation order increases. Finally, comparing with the benchmark MMSE method, the proposed ones can significantly reduce computational complexity while achieving satisfactory SER performance.

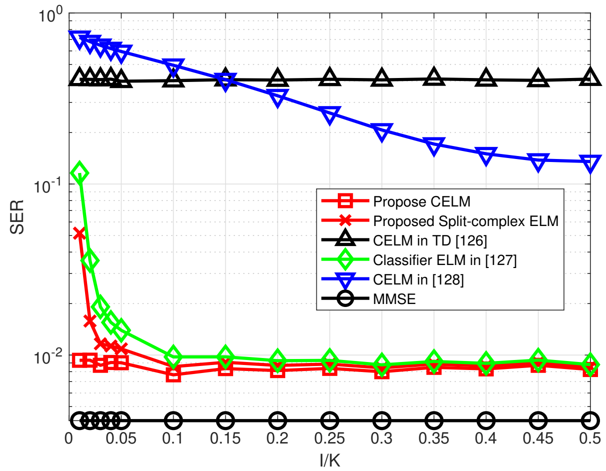

Figure 4.5: SER against various training/testing ratio I/K.

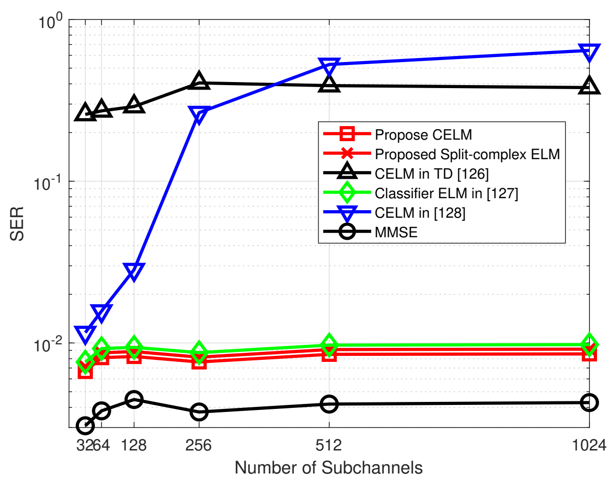

Figure 4.6: SER against various subchannels.

Chapter 5

Pressure Wave Analysis Based Pipeline Leak Detection by OPELM and BiLSTM*

5.1 Introduction

Pipeline leak detection has been extensively studied for years due to its crucial importance in oil and gas industry. According to statistics [129], most of the products are transported through pipeline networks which benefit from its lower overhead cost and greater transportation capability, compared to other transportation methods. However, the rupture or leak issue may incur huge property loss and environmental hazard especially when the leak spot lacks monitoring. Hence, various techniques for pipeline leak detection and localization have been proposed. Pressure monitoring based leak detection is a commonly used internal method that continuously performs pressure point analysis (PPA) to detect leak events. The PPA based approach owns several advantages [130, 131], such as fast response, high sensitivity, continuous monitoring, and easy installation/maintenance.

Traditionally, model based PPA methods are dominant. In [132], a modified model analysis method is proposed to analyze the transient process for leak detection and localization. In [133], the detection method combining

*A version of this chapter has been published as "A novel PPA method for fluid pipeline leak detection based on OPELM and bidirectional LSTM," in *IEEE Access*.

pipeline dynamic model and extended Kalman filter is proposed by which the detection accuracy is greatly improved. However, these model based methods require prior model knowledge, and have limited accuracy and flexibility in noisy and complex industrial situations.

In recent years, ML and data-driven techniques have been widely accepted and employed in industries with resources such as abundant sensors and big data. Successful applications of ML have been reported in plant wide system/process monitoring and fault diagnosis, [134, 135]. Although pipeline leak monitoring is distributed in nature, traditional leak detection techniques such as PPA usually involves local and point-wise inspections. Moreover, recent research has shown promising applications of data-driven ML techniques in pipeline systems, which can not only achieve rapid and reliable local leak detection, but also be extended to designing the distributed pipeline health monitoring system. In [136, 137], the application of back-propagation neural networks (BPNN) in leak detection is experimented and discussed. As one of the commonly applied ML methods, SVM is also implemented in leak detection as shown in [138, 139]. In [140, 141], k-nearest neighbor (KNN) algorithm is adopted for pipeline leak detection and rupture size estimation. Its performance is validated through comparison study with some other ML methods. ELM which owns the merits of good approximation ability and fast learning speed, has been applied in leak detection in [142, 143]. It is shown that the time spent on model learning is greatly reduced. Applications of other ML methods in leak detection are also reported, including naive Bayesian (NB) based [144] and decision tree (DT) based classifier [145]. Comparing to the analytical model based method, the ML based ones can improve the detection accuracy and generalization performance when adapting to different industrial situations.

However, one challenge for deploying PPA based methods is the high rate of false alarms [146]. This is due to the fact that frequent pump or valve manipulations may also lead to pressure drops, which can be mistakenly detected as leaks. Due to such a drawback, PPA is usually taken as a supplement to other leak detection methods [147], adding complexity to a practical leak detection strategy. Thus, false alarm elimination becomes crucial for deploying PPA based methods in practice. In [148] and [149], the flow balance method is employed to assist the discrimination of false alarms through installation of flow meters at investigated points. In [150], a multi-sensor paring method is proposed to decide the genuineness of a leak by considering the feasibility of paired pressure drop time instants. The methods in [148–150] are conditioned on the prior knowledge such as multiple flow readings and installation topology of pressure sensors. Furthermore, a pattern matching method which compares the similarity of pressure drops between the real leak and normal adjustments is also proposed. In [151], a two-stage decision scheme is presented where the short-term and long-term models are trained respectively and a switching threshold is set to decide the proper model. This method intends to utilize the better fitting model to perform detection with respect to different lengths of pressure sequences, such that the number of false alarms can be reduced. Experiments show that the detection accuracy is improved while false alarms decrease if the model can be correctly selected. However, the appropriate value of the switching point for choosing suitable model is difficult to obtain which hinders its practical implementation.

In view of these drawbacks, in this chapter a machine learning based PPA method is proposed which can perform accurate leak detection with significantly reduced false alarms. The proposed method is based on supervised OPELM combined with BiLSTM networks, which is shown to improve the performance and enhance the practicality of the pressure monitoring based leak detection. Main contributions of this chapter are described as follows:

- An effective PPA leak detection method based on supervised OPELM combining BiLSTM is proposed to achieve higher detection accuracy and significantly less false alarms, compared to existing ML based PPA methods.

- The strong past and future memorizing ability of BiLSTM is firstly utilized to identify the true and false leaks by considering the ambient pressure status around the suspicious leaks.

- Several unique characterizations of leak features are proposed and the effectiveness is verified through experiments.

- Performance of the proposed method is assessed and compared with various ML based methods through multiple experiments on different industrial data sets.

The remainder of this chapter is organized as follows. Section 5.2 presents the main leak detection methodology. Section 5.3 includes the experiment and comparison results with discussions. Finally, conclusion is drawn in Section 5.4.

5.2 Main Methodology

5.2.1 Motivation of The Proposed Method

Although the high sensitivity to pressure changes owned by PPA can lead to fast leak detection response, it may also contribute to the high number of false alarms. Fig. 5.1 is given as an example to show how a false alarm can occur by closely comparing the real leak wave and the normal pressure fluctuating wave (that may be taken as a leak by mistake). The top plot in Fig. 5.1 displays the process of a real leak experiment where the red colored parts are corresponding to leak events. The bottom plot in Fig. 5.1 displays a process of non-leak normal working pressure fluctuation.

The current PPA based methods usually take the typical leaking pressure transient as a signature (e.g., the $2nd$ red colored portion from left in the top of Fig. 5.1) to detect leaks. When a section of pressure wave is deemed similar to the signature, it may be considered as a suspicious leak. For example, the wave sections between the narrow red dashed lines in the bottom of Fig. 5.1 may be considered as leaks because they bear certain similarity with the leak signature in the top plot. Thus, when applying the conventional PPA based method, many false alarms may be generated due to the existence of numerous similar transient pressure drops in the normal working process.

However, given the typical leak pressure wave shown in the top of Fig. 5.1, it is easy for the human to tell that the bottom plot does not show an actual leak process. This is because in human perception, both of the pressure drop contained in the narrow window (the gap between red dashed lines), and its ambient pressure in the wider temporal window (such as the gap between blue dashed lines) are under consideration. In other words, Fig. 5.1 shows similar sudden pressure drops in both leak and non-leak processes, however, if we observe a wider temporal range, their ambient pressure characteristics are obviously different.

If we merely choose a wider leak signature (i.e., expand the red-colored leak portion to a wider range), then the most prominent leak feature, usually manifested as a sudden pressure drop, may become indistinct due to the mixing of the ambient pressure status. Hence, a method that can imitate the above human perception is desirable, which can quickly capture a possible leak, and then pinpoint a true leak by excluding false alarms using ambient pressure information. The former can be achieved by proper feature extraction and classification, while the latter can be realized by BiLSTM, which has strong memorizing ability to treat the ambient pressure information.

Consequently, we propose a two-stage PPA leak detection scheme based on combined OPELM and BiLSTM networks. It exploits the fast learning and superior classification performance of OPELM to perform first-stage detection and then takes the strong memorizing advantage of BiLSTM to broaden the temporal observation range, thus effectively eliminates false alarms.

5.2.2 Structure of The Proposed Leak Detector

For the pipeline leak detection studied in this chapter, the training and detection phases of the proposed method are shown in Fig. 5.2 and Fig. 5.3.

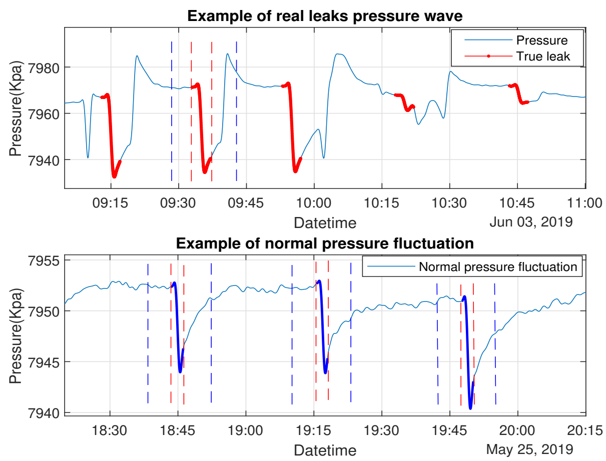

Figure 5.1: Example of true leaks and normal pressure fluctuation.

Since the method is combining OPELM and BiLSTM, both the training and detection processes can be conducted in two stages as marked in Fig. 5.2 and Fig. 5.3. In the first stage of training process, raw labeled pressure data P_r^t is passed to a low-pass filter (LPF) to remove high frequency noises. The superscript t in P_r^t represents the training process and the subscript r indicates raw pressure data. The output of LPF is the filtered pressure data, denoted as P_f^t. From P_f^t, the leak portion matrix $\boldsymbol{P}_l^t$ and non-leak portion matrix $\boldsymbol{P}_{nl}^t$ can be retrieved in the same length m according to the known label information. Apply feature extraction which is to be given in Section 5.2.4 on $\boldsymbol{P}_l^t$ and $\boldsymbol{P}_{nl}^t$ to obtain feature matrix for training, denoted as $\boldsymbol{F}_{tr}$, and feature matrix for testing, denoted as $\boldsymbol{F}_{te}$, for which the corresponding labels are denoted as Y_{tr} and Y_{te}, respectively. Thus, the OPELM network can be obtained through training.

In the second stage of training process, the testing result $\hat{Y}_{te}^t$ from the first stage, which contains suspicious leaks, is separated into true positive (TP) and false positive (FP) groups. The time instant vectors of the suspicious leaks T_{TP} and T_{FP} are taken as centers of the training sequences for BiLSTM. The training sequences of TP and FP, denoted as $\boldsymbol{S}_{TP}^t$ and $\boldsymbol{S}_{FP}^t$, respectively, are taken from $\boldsymbol{F}^t$, the feature matrix of the entire training pressure data. Thus, the training feature set for BiLSTM is established. After performing BiLSTM training, both the OPELM and BiLSTM networks are obtained.

Once OPELM and BiLSTM networks are constructed, leak detection can be performed as shown in Fig. 5.3. Denote the raw pressure data as P_r^d, where the superscript d represents the detection process. In the first stage of detection process, P_r^d is passed through LPF, feature extraction and OPELM detection. The output $\hat{Y}_{elm}^d$ is then fed to the second stage to further discriminate the TPs and FPs. The input sequences of BiLSTM, $\boldsymbol{S}^d$, are centered according to the suspicious leak time instants in $\hat{Y}_{elm}^d$ and selected from the extracted features in $\boldsymbol{F}^d$. Then, the output of BiLSTM are the final detection result $\hat{Y}_{lstm}^d$.

Details of the proposed method are further elaborated in the following subsections.

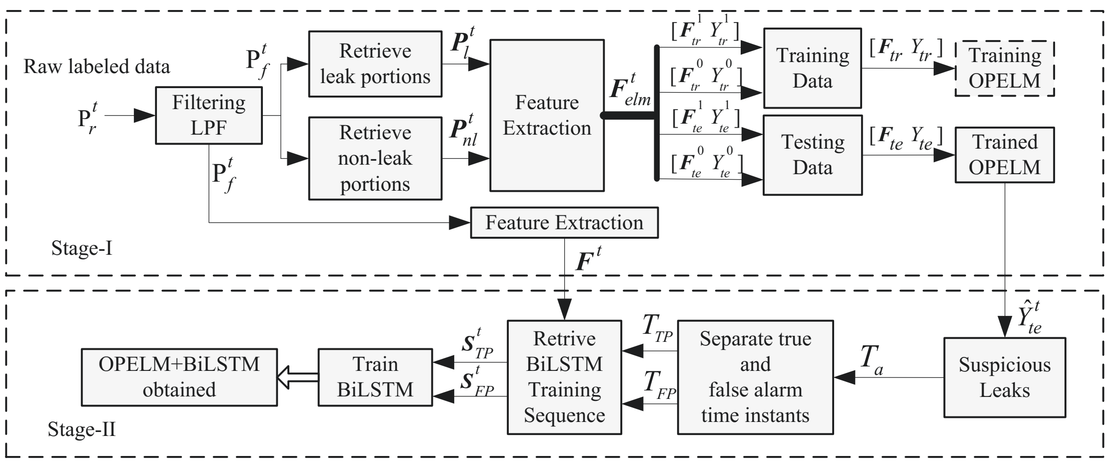

Figure 5.2: Block diagram of the training process.

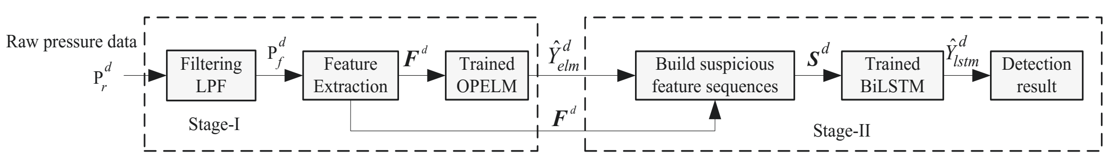

Figure 5.3: Block diagram of detection process.

5.2.3 Data Pre-processing

There are two main steps in the data pre-process, i.e., data filtering, data sectioning and labeling.

1. Data filtering: the raw training pressure data P_r^t is from the records of leak experiments and is therefore well labeled. It contains high frequency noises which may jeopardize the accuracy of leak detection, thus, LPF is applied to eliminate the noise. In this chapter, the low pass FIR filter of order 20 is employed with the sampling frequency $1Hz$ and the cutoff frequency of $0.1Hz$. Although the filter structure and parameters may vary from case to case depending on the noise condition, the filtered data should retain the characteristics of transient leak pressure drops. It can be observed in Fig. 5.4 that after filtering, the pressure wave is smoother with excessive noises eliminated.

2. Data sectioning and labeling: After raw data is filtered, to establish the training sets for OPELM, the typical leak portion matrix $\boldsymbol{P}_l^t$ and non-leak portion matrix $\boldsymbol{P}_{nl}^t$ are singled out from P_f^t. The $\boldsymbol{P}_l^t$ can be sectioned from P_f^t according to the provided leak information. For example, as shown in the bottom plot in Fig. 5.4, there are 9 leak portions retrieved from the data of Site-1. The length of the leak portion is set to be m seconds where m may vary with respect to different leak situations. In the case shown in Fig. 5.4, m is set to be $240s$ which appropriately covers the pressure impact duration caused by the rupture. The size of $\boldsymbol{P}_l^t$ is $r \times m$, where r is the number of leak portions. It can be observed that the negative pressure impacts caused by leaks usually have similarities in shape and dropping trend, but the amplitude may vary due to different leaking conditions such as leak size, pipeline pressure status, flow rate and pipe contents, etc. Each row of the non-leak pressure matrix $\boldsymbol{P}_{nl}^t$ is randomly sectioned from P_f^t by avoiding the leak instants. Assume q non-leak portions are selected, the size of $\boldsymbol{P}_{nl}^t$ is $q \times m$. The length of a non-leak portion is also m seconds.

5.2.4 Features Extraction

A feature extraction scheme is proposed to represent leak characteristics, as shown in Fig. 5.5. For each pressure portion, 5 features are extracted including "Similarity", "Interception", "Slope rate", "Area" and "Variance". Thus, the size of the extracted feature matrix with n pressure portions is $n \times 5$.

Denote the feature matrix for training OPELM as $\boldsymbol{F}^t_{elm}$, then, it is divided into the training and testing parts by K-fold training scheme, written as $\boldsymbol{F}^t_{elm} = [\boldsymbol{F}^T_{tr}, \boldsymbol{F}^T_{te}]^T$. Combining the label vectors, the labeled training feature matrix can be written as $[\boldsymbol{F}_{tr}\ Y_{tr}]$, where $\boldsymbol{F}_{tr} = [\boldsymbol{F}^{1\,T}_{tr}\ \ \boldsymbol{F}^{0\,T}_{tr}]^T$ and $Y_{tr} = [Y^{1T}_{tr}\ Y^{0T}_{tr}]^T$. The superscript 1 and 0 are referring to the class 1 (leak) and class 0 (non-leak), respectively, and Y^1_{tr}, Y^0_{tr} are the corresponding label vectors. $\boldsymbol{F}_{te}$ and Y_{te}, are defined similarly except that they are for testing the trained OPELM classifier.

In the following, the characterization of the leak signature and several features are calculated. It should be noted that the "Similarity" related calculations in 5.2.4 and 5.2.4 are performed on normalized $\boldsymbol{P}^t_l$ and $\boldsymbol{P}^t_{nl}$ with range of $[-1, 1]$.

1) Leak signature characterization

As can be viewed in Fig. 5.4, multiple leak portions are retrieved from P^t_f, some of them vary in amplitudes and shapes. Therefore, the one which is the most similar to other leak portions but dissimilar to the non-leak portions should be chosen as the leak signature. The leak signature selection process is introduced as follows.

Assume one of the r normalized leak portions is written as $P^t_{li} \in \boldsymbol{P}^t_l$, $i = 1 \cdots r$, $P^t_{li} = [p_{i1} \cdots p_{im}]$ and $\boldsymbol{P}^t_l$ is a $r \times m$ matrix. Similarly assume one of the q normalized non-leak portion is written as $P^t_{nlj} \in \boldsymbol{P}^t_{nl}$, $j = 1 \cdots q$, $P^t_{nlj} = [p_{j1} \cdots p_{jm}]$ and $\boldsymbol{P}^t_{nl}$ is sized at $q \times m$. The subscripts "l" and "nl" are corresponding to the "leak" and "non-leak" respectively. Define the total number of selected leak and non-leak portions as $n = r + q$.

Apply exponential function to calculate the similarity matrix $\boldsymbol{W}$ of the concatenated matrix $\boldsymbol{P}^t = \left[\boldsymbol{P}_l^{tT}, \boldsymbol{P}_{nl}^{tT}\right]^T$, where $\boldsymbol{P}^t$ is sized at $n \times m$. The element w_{ij} representing the similarity between the i-th row P_i^t and the j-th row P_j^t in $\boldsymbol{P}^t$ is written as,

$$w_{ij} = e^{-\frac{||P_i^t - P_j^t||}{2}} \qquad i, j = 1 \cdots n. \tag{5.1}$$

Thus, the similarity matrix $\boldsymbol{W}$ of $\boldsymbol{P}^t$ is obtained as,

$$\boldsymbol{W} = \begin{bmatrix} w_{11} \cdots w_{1r} \cdots w_{1n} \\ \vdots \ \cdots \ \vdots \ \cdots \ \vdots \\ w_{i1} \cdots w_{ir} \cdots w_{in} \\ \vdots \ \cdots \ \vdots \ \cdots \ \vdots \\ w_{n1} \cdots w_{nr} \cdots w_{nn} \end{bmatrix} \tag{5.2}$$

When the Euclidean distance between P_i^t and P_j^t is small, the value of w_{ij} is close to 1; Oppositely, if the Euclidean distance is big which means the two vectors are obviously different, the value of w_{ij} is approaching 0.

The leak signature is chosen as the i-th leak portion P_{li}^t in $\boldsymbol{P}_l^t$ from the following

$$\underset{i}{\operatorname{argmax}} \left(\sum_{j=1}^{r} w_{ij} - \sum_{j=r+1}^{n} w_{ij} \right), \ i \in [1 \cdots r]. \tag{5.3}$$

The first sum in (5.3) indicates the similarity of i-th leak portion with other leak portions including itself, and the second sum indicates the similarity of the i-th leak with the non-leak portions. It is to choose the leak portion which has the highest similarity sum value with other leak portions and lowest similarity sum value with non-leak portions. Denote the chosen leak signature P_{li}^t as the template P_s^t, where the subscript s means signature.

2) Similarity

Take P_s^t as the template, the feature f_1 representing similarity is calculated by (5.4). As shown in Fig. 5.5, the "similarity" is calculated between a normalized pressure portion P_i^t in green and the chosen template P_s^t in blue.

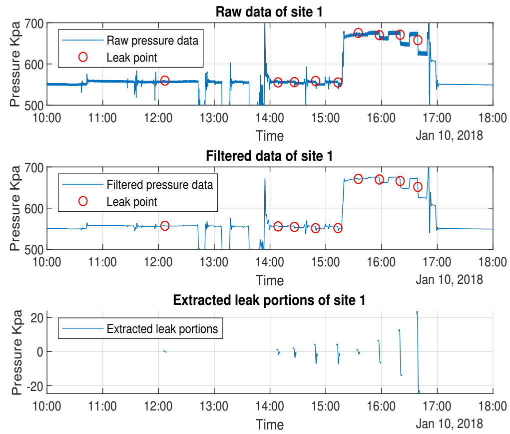

Figure 5.4: Filtering and leak portions of Site-1.

Given a pressure portion $P_i^t \in \boldsymbol{P}^t, i = [1, \cdots, n]$, the similarity feature of the i-th pressure portion is obtained as:

$$f_{i1} = e^{-\frac{||P_i^t - P_s^t||}{2}} \tag{5.4}$$

Although f_{i1} can reflect the similarity between the normalized pressure vectors and the chosen template, it still has drawbacks. For example, when the amplitude of a pressure vector is obviously different to the template, after normalization, the difference is ignored due to the consistent normalizing range $[-1, 1]$. It may result in high similarity value in f_{i1} as long as the two shapes are alike after normalization. In view of this drawback, other factors are also considered.

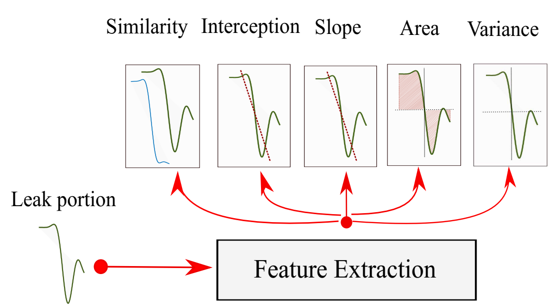

Figure 5.5: Diagram of feature extraction.

3) Slope rate and interception

The feature "similarity" represents the overall shape similarity after normalization. However, the dynamic characteristics such as "slope rate" and "interception" which reflects the intensity of a pressure dropping also need to be considered. Instead of implementing normalization, the pressure portion is performed with mean centering.

Apply minimum mean squared error (MMSE) based linear fitting on the selected portion P_i^t after mean centering, the two features can be found as:

$$[\alpha, \beta] = \underset{\alpha,\beta}{\operatorname{argmin}} MSE(P_i^t - \hat{P}_i^t) \tag{5.5}$$

$$\hat{P}_i^t = \alpha t + \beta \tag{5.6}$$

where $\hat{P}_i^t$ is the linear fitted vector of P_i^t when slope and interception are α and β respectively.

Features of slope rate and interception for P_i^t can be written as:

$$f_{i2} = \alpha \tag{5.7}$$

$$f_{i3} = \beta \tag{5.8}$$

4) Area

The factor representing the feature of amplitude is considered in f_{i4} by calculating the area. The amplitude indicating the distance between the maximal and minimal values is a commonly used feature to evaluate the pressure drop. This feature may be effective when the pressure wave is smooth and no outlier exists. However, the sharp spikes or outliers usually exist, hence the amplitude may not faithfully represent the pressure drop. It is necessary to use an alternate measure which is less sensitive to spikes and outliers to represent the essential pressure drop. For the selected data vector of length m, after mean centering, the area formed by non-normalized pressure portion P_i^t and time axis is calculated by discrete integration method, which is always positive and not sensitive to outliers. So the "Area" is taken as the 4th feature to represent the general pressure drop, it can be approximately written as follows.

$$f_{i4} = \sum_{j=1}^{m} |p_j - \mu| \times \Delta t, \qquad p_j \in P_i^t \tag{5.9}$$

where μ and Δt are the mean value of P_i^t and sampling time interval respectively.

5) Variance

The variance representing the extent of variation of a non-normalized pressure portion P_i^t is also taken as a feature, which is calculated as:

$$f_{i5} = \frac{1}{m} \sum_{j=1}^{m} (p_j - \mu)^2, \qquad p_j \in P_i^t \tag{5.10}$$

6) Concatenation of features

After the features for class 1 (Leak) and class 0 (Non-leak) are extracted, the training set for OPELM can be established by concatenation of $\boldsymbol{F}_{tr}^0$ and $\boldsymbol{F}_{tr}^1$, the corresponding label values are Y_{tr}^1 and Y_{tr}^0, where

$$\boldsymbol{F}_{tr}^1 = \begin{bmatrix} f_{11}^1 \cdots f_{15}^1 \\ \vdots \;\cdots\; \vdots \\ f_{r1}^1 \cdots f_{r5}^1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1^{1T} \\ \vdots \\ \boldsymbol{f}_r^{1T} \end{bmatrix} Y_{tr}^1 = \begin{bmatrix} +1 \\ \vdots \\ +1 \end{bmatrix} = \begin{bmatrix} y_1^1 \\ \vdots \\ y_r^1 \end{bmatrix} \tag{5.11}$$

$$\boldsymbol{F}_{tr}^0 = \begin{bmatrix} f_{11}^0 \cdots f_{15}^0 \\ \vdots \;\cdots\; \vdots \\ f_{q1}^0 \cdots f_{q5}^0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1^{0T} \\ \vdots \\ \boldsymbol{f}_q^{0T} \end{bmatrix} Y_{tr}^0 = \begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix} = \begin{bmatrix} y_1^0 \\ \vdots \\ y_q^0 \end{bmatrix} \tag{5.12}$$

However, in practice, the data acquired during leak events are very rare and valuable. It is usually difficult to have enough class 1 training samples from real leak events. In this case, a training sample generation scheme is applied.

Assume there are l real leak events, and the r training vectors of class 1 are generated based on the l real leak vectors. The training features are generated following a Gaussian distribution whose mean values are corresponding to the

l real leak feature vectors and the user defined standard deviations. Theoretically, if more real leak events can be recorded to analyze the distribution of features, the feature generation scheme may achieve better approximation by following the analyzed distribution instead of Gaussian. On the other hand, the original data for training class 0 is plenty and in our case the q samples are randomly selected.

7) Distribution of extracted features

The feature extraction maps the pressure data to a higher dimensional feature space to enhance the feature representation. As shown in Fig. 5.6, the distribution of extracted features can be easily classified. The three axes are chosen as the first three principle components (PCs) from the principal component analysis (PCA) method. The separable leak and non-leak distributions and experiment results given in Section 5.3 verify the effectiveness of proposed feature extraction scheme.

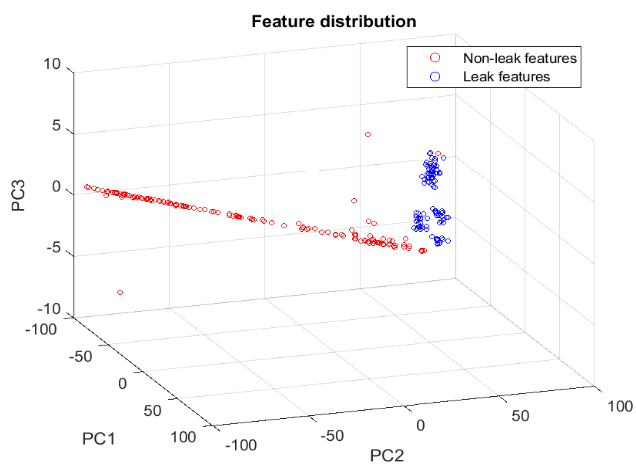

Figure 5.6: Distribution of feature extraction.

5.2.5 The First-stage OPELM Training

1) ELM structure

OPELM is a variant of ELM which can optimally prune the number of hidden neurons. It shares the same structure with ELM, which is depicted in

Fig. 5.7 with details given as follows.

- Input layer: it contains 5 nodes corresponding to the 5 elements of the i-th extracted feature vector $\boldsymbol{f}_i = [f_{i1}, \cdots, f_{i5}]^T$, where $i \in [1, n]$.

- Hidden layer: it contains L hidden nodes with activation function $g(x) : R^5 \rightarrow R$. The output of the l-th hidden node is given by

$$h_l = g(\mathbf{a}_l{}^T \cdot \boldsymbol{f}_i + b_l), \tag{5.13}$$

where $l = 1, \cdots, L$ and $\mathbf{a}_l$ is the 5×1 column vector of the weights connecting the l-th hidden node and the 5 input nodes. b_l is the bias of the l-th hidden node. Here $\mathbf{a}_l$ and b_l are randomly assigned following the uniform distribution over $[-0.2 \quad 0.2]$ and kept fixed afterwards.

- Output layer: it has only 1 node corresponding to the label y_i when input is $\boldsymbol{f}_i$. Thus, the output of an ELM network with L hidden nodes can be written by

$$y_i(\boldsymbol{f}_i) = \sum_{l=1}^{L} h_l \boldsymbol{\beta}_l = \boldsymbol{h}_i \boldsymbol{B}, \quad \boldsymbol{\beta}_l \in R, \tag{5.14}$$

where $\boldsymbol{h}_i$ is a row vector of the L hidden nodes outputs. The output weight matrix $\boldsymbol{B}$ includes L output weights connecting the hidden layer and output layer. It can be written as $\boldsymbol{B} = [\boldsymbol{\beta}_1 \cdots \boldsymbol{\beta}_L]^T$, and can be obtained by training the network, which is explained in the following.

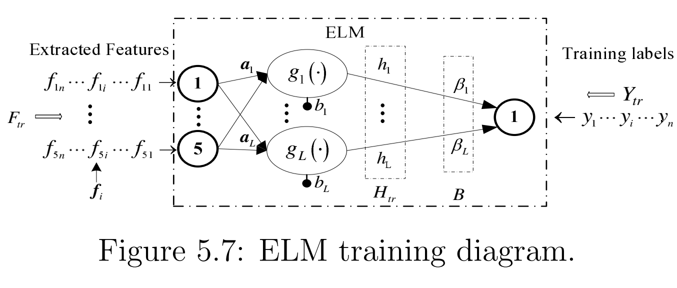

Figure 5.7: ELM training diagram.

2) ELM training

To train ELM network to obtain the output weight matrix $\boldsymbol{B}$, the extracted features in Section.5.2.4 and their label vectors are used as training samples, which are denoted as $\boldsymbol{F}_{tr} = [{\boldsymbol{F}_{tr}^{1}}^{T}, {\boldsymbol{F}_{tr}^{0}}^{T}]^{T} = [\boldsymbol{f}_1^1, \cdots, \boldsymbol{f}_r^1, \boldsymbol{f}_1^0, \cdots, \boldsymbol{f}_q^0]^T$, and $Y_{tr} = [{Y_{tr}^{1}}^{T}, {Y_{tr}^{0}}^{T}]^{T} = [y_1^1, \cdots, y_r^1, y_1^0, \cdots, y_q^0]^T$, where $n = r + q$.

Let

$$\boldsymbol{H}_{tr} = \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_i \\ \vdots \\ \boldsymbol{h}_n \end{bmatrix} = \begin{bmatrix} g(\mathbf{a}_1^T \boldsymbol{f}_1 + b_1) \cdots g(\mathbf{a}_L^T \boldsymbol{f}_1 + b_L) \\ \vdots \qquad \cdots \qquad \vdots \\ g(\mathbf{a}_1^T \boldsymbol{f}_i + b_1) \cdots g(\mathbf{a}_L^T \boldsymbol{f}_i + b_L) \\ \vdots \qquad \cdots \qquad \vdots \\ g(\mathbf{a}_1^T \boldsymbol{f}_n + b_1) \cdots g(\mathbf{a}_L^T \boldsymbol{f}_n + b_L) \end{bmatrix} \tag{5.15}$$

$\boldsymbol{H}_{tr}$ denotes the $n \times L$ hidden layer outputs matrix in the training phase and $\boldsymbol{f}_i$ is the ith column of $\boldsymbol{F}_{tr}$. The n rows of $\boldsymbol{H}_{tr}$ are the hidden layer output vectors corresponding to the n input feature vectors during training.

Establish the cost function considering the empirical and structural risk of the proposed ELM and obtain the following:

$$Min : \mathcal{L}_{ELM} = \frac{1}{2}||\boldsymbol{B}||^2 + \frac{C}{2}||\boldsymbol{Y}_{tr} - \boldsymbol{H}_{tr}\boldsymbol{B}||^2 \tag{5.16}$$

Solving the regularized least squares optimization problem in (5.16), we have

$$\boldsymbol{B} = \left(\boldsymbol{H}_{tr}^T \boldsymbol{H}_{tr} + \frac{\boldsymbol{I}_L}{C} \right)^{-1} \boldsymbol{H}_{tr}^T \boldsymbol{Y}_{tr}, \tag{5.17}$$

where C is the coefficient for balancing the empirical and structural risks, $\mathbf{I}_L$ is the L dimensional identity matrix.

In a nutshell, the overall procedures of ELM establishment part and optimal-pruning part are described separately as in Algorithm 1 and Algorithm 2.

5.2.6 The Second-stage BiLSTM Network

The objective of BiLSTM is to further identify false alarms (FP) and true alarms (TP). As the raw data is already labeled during the first-stage training, the feature sequences centered with true leak instants are chosen as TP training sequence. On the other hand, the feature sequences centered with the false alarm instants obtained in OPELM are taken as FP training sequences. Thus, the BiLSTM can be trained to flag false alarms. Fig. 5.8 gives an example of training sequences of TP and FP respectively. BiLSTM can memorize the characteristics over time for the two sets of sequential features, which can then be classified correspondingly. The details can be found as follows.

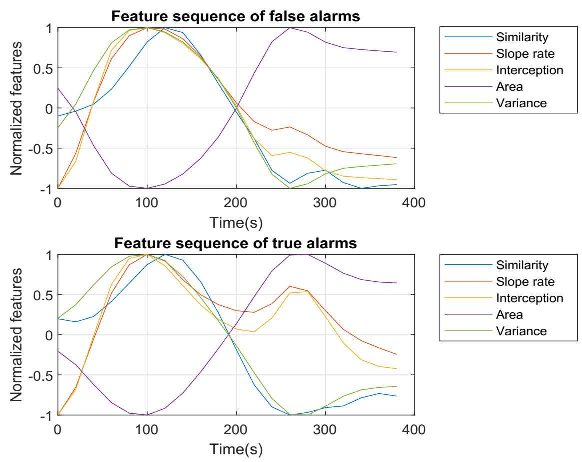

Figure 5.8: Example of BiLSTM training sequence.

1) BiLSTM training sequences selection and training

The time span of a BiLSTM memory depends on the temporal length of the training sequence. Define the length of BiLSTM training sequence as s, then the memory time of a BiLSTM is $(s-1)\times\delta$, where δ is the step size of the selected section of pressure wave. With $s=20$ and the step size $\delta=30s$, the pressure variation characteristics within the time span of 570 seconds can be memorized by the BiLSTM to identify if a reported alarm is true or false.

As shown in Fig. 5.2, the sequences for training BiLSTM are chosen from the extracted feature $\boldsymbol{F}^t$ according to the first-stage testing result $\hat{Y}^t_{te}$, and $\boldsymbol{F}^t$ is obtained from the entire labeled pressure data set. Assume the total length of labeled pressure wave as N, then there will be $K=\lfloor(N-m)/\delta)\rfloor$ sectioned portions, where $\lfloor\cdot\rfloor$ takes the nearest lower integer. Hence, the corresponding feature matrix $\boldsymbol{F}^t$ has the dimension of $K \times 5$, the median time instants of sectioned portions are denoted as T, where $T = [t_1, \cdots, t_K]$.

The suspicious alarm time instants, denoted as T_a, are selected from T according to the first-stage result $\hat{Y}_{te}^t$. T_a contains the time instants with the corresponding test result in $\hat{Y}_{te}^t$ are in class 1. Since the label information is known, the suspicious alarm time instants T_a can be separated as true positive T_{TP} and false positive T_{FP}. Take the instants in T_{TP} and T_{FP} as centers, choose sequences from $\boldsymbol{F}^t$ with length s, the BiLSTM training sequences for true positive $\boldsymbol{S}_{TP}^t$ and false positive $\boldsymbol{S}_{FP}^t$ are established.

The training process of BiLSTM is usually performed by back-propagation through time (BPTT) algorithm. The overall procedures of the BiLSTM training is summarized in Algorithm 5.

Algorithm 5 Second-stage BiLSTM training process.

1: Extract the time instants of the true and false alarms T_{TP} and T_{FP} according to the first-stage test result $\hat{Y}_{te}^t$;
2: Retrieve the feature sequences $\boldsymbol{S}_{TP}^t$ and $\boldsymbol{S}_{FP}^t$ from $\boldsymbol{F}^t$, the center time instants of retrieved sequences are T_{TP} and T_{FP}, and the length of retrieved sequences is s;
3: Normalize $\boldsymbol{S}_{TP}^t$ and $\boldsymbol{S}_{FP}^t$ within $[-1, 1]$ and set BiLSTM structural parameters.
4: Train the BiLSTM with normalized $\boldsymbol{S}_{TP}^t$ and $\boldsymbol{S}_{FP}^t$ as a classifier.

5.2.7 Detection

After the OPELM and BiLSTM networks are trained, the detection process can be applied as shown in Fig. 5.3. It can be separated into two stages similarly as the training process.

1) First-stage OPELM detection

In the first-stage of detection, the raw pressure data P_r^d is fed to LPF to remove high frequency noise and obtain the filtered pressure data P_f^d, where the superscript d represents the detection process. Then, the filtered pressure data is transformed to feature space denoted as $\boldsymbol{F}^d$ via the feature extraction procedure, given in Section.5.2.4. $\boldsymbol{F}^d$ is then passed to the trained OPELM networks to obtain the first-stage detection result $\hat{Y}^d_{elm}$.

2) Second-stage BiLSTM detection

Provided with $\hat{Y}^d_{elm}$ from the previous stage, the feature sequences for the second-stage detection can be constructed following the same procedure as in the training phase. Take the sequences according to the suspicious leak instants in $\hat{Y}^d_{elm}$ as input, denoted as $\boldsymbol{S}^d$, the output of BiLSTM networks $\hat{Y}^d_{lstm}$ is the final detection result.

5.3 Experiment

In this section, the proposed two-stage PPA leak detection method is validated on data sets collected from leak experiments on four different industrial sites. In addition, a thorough comparison study is performed to verify the effectiveness of the proposed method. The raw data is measured by the pressure sensors installed along the pipelines. The first pipeline's content is oil/gas while the other three pipelines contain salt water. Besides, the pressure fluctuation on Site-1 and Site-2 are less than that on Site-3 and Site-4.

5.3.1 Evaluation Criteria

To evaluate the detection performance, some of the commonly used measures are employed. Denote the counts of positive events (i.e., leaks) as P and negative events (i.e., normal operations) as N. From the detection result, denote the counts of true positive as TP, true negative as TN, false positive as FP and false negative as FN.

- TPR: true positive rate which is the percentage of leaks that are correctly detected.

$$TPR = \frac{TP}{P} \times 100\% \tag{5.18}$$

- FDR: false discovery rate which indicates the ratio between false positives and all the detected positives. It can be calculated as the the percentage of false leak alarms in all the leak alarms.

$$FDR = \frac{FP}{TP + FP} \times 100\% \tag{5.19}$$

- ACC: the detection accuracy which indicates the ratio between the number of correct detection counts and all events. It is used to assess the overall detection accuracy including leak events and normal operation events.

$$ACC = \frac{TP + TN}{P + N} \times 100\% \tag{5.20}$$

5.3.2 Results

The proposed OPELM+BiLSTM method employs single hidden layer feed-forward neural network structure for OPELM, and 5-layer structure for BiLSTM, which are 'Sequence input layer', 'BiLSTM layer', 'Fully connected layer', 'Softmax layer' and 'Classification output layer'. The corresponding parameters and their values are listed in Table.5.1.

The overall leak detection results on 4 industrial sites are summarized in Table. 5.2. The results displayed are averaged over 100 detection results. From TPR which represents the percentage of detected leaks among all leak events, it can be seen that most of the leak events are successfully detected. Furthermore, by comparing the FP results of the two stages, it shows that the second stage can greatly decreases the number of FP. Moreover, it should be noted that the detection performance is related to the choice of m, the length of pressure template. The appropriate value of m is dependent on multiple factors regarding the intensity of pressure fluctuations. The relation between detection performance and m is further investigated and discussed in subsection 5.3.5 and the results listed in Table.5.2 are based on the chosen values.

More detailed detection results for each site are given as follows.

Table 5.1: Related parameters of proposed method.

Model	*Parameter*	*Value*	*Parameter*	*Value*
OPELM	Input Nodes	5	Kernel	Gaussian
	Hidden Nodes	31	Output Nodes	1
BiLSTM	Input Size	5	Output Size	2
	Hidden Units	200	Gate Activation	*Sigmoid*
	State Activation	*tanh*	Softmax Layer	*Softmax*

Table 5.2: Experiment results on different industrial sites.

Site	*Stage*	*TP*	*FP*	*P*	*N*	*TPR*	*FDR*	*ACC*	*m*
1	1	8.7	426.4	9	71698	96.6%	98%	99.42%	120*s*
	2	8.5	5	9	71698	94.44%	37.04%	99.99%	
2	1	9.7	92.4	10	116150	97%	90.49%	99.92%	160*s*
	2	9	2.9	10	116150	90%	24.37%	99.99%	
3	1	15.2	409	18	120272	84.44%	96.42%	99.66%	300*s*
	2	15.1	5.9	18	120272	83.89%	28.09%	99.99%	
4	1	17.9	261.1	19	120655	94.21%	93.58%	99.78%	260*s*
	2	17.6	11.4	19	120655	92.63%	39.31%	99.98%	

1) Experiment on data from Site-1

The leak events description of Site-1 is listed in Table.5.3. There are 9 leak events experimented on an oil and gas pipeline at different time instants. All the 9 leaks are successfully detected even for the small leak in the first event. Details of the first and second stage results are depicted in Fig. 5.9.

- First stage result of Site-1: The top plot in Fig. 5.9 shows the overview of the first-stage detection result. It can be observed that many suspicious leaks are detected. There are totally 435 alarms in the first-stage result where only 9 of them are TP. Considering the number of total events is over 71700, even with the number of 426 false alarms, the *Accuracy* (ACC) is still higher than 99%. However, more than 400 false alarms may be troublesome in practice.

- Second stage result of Site-1: The middle and bottom plots in Fig. 5.9

Table 5.3: Experiment environment of Site-1.

Index	*Pipe Flow*	*Leak Flow*	*Leak Percentage*	*Detected*
1	$100m^3/day$	$1.44m^3/day$	1.44%	Yes
2	$100m^3/day$	$5.74m^3/day$	5.74%	Yes
3	$100m^3/day$	$10.25m^3/day$	10.25%	Yes
4	$100m^3/day$	$20.04m^3/day$	20%	Yes
5	$100m^3/day$	$20.04^3/day$	20%	Yes
6	$300m^3/day$	$3m^3/day$	1%	Yes
7	$300m^3/day$	$15.26m^3/day$	5.08%	Yes
8	$300m^3/day$	$1.25m^3/day$	10%	Yes
9	$300m^3/day$	$2.5m^3/day$	20%	Yes
Data Length: 399 *hours* Leak Event: 9 Content: Oil, Gas				

demonstrate the second-stage detection result of Site-1. It shows that the alarms number has been dramatically decreased from 435 to 13 where the 9 cases of TP are all detected and the false ones are mostly eliminated. Referring to Table.5.3, even the $1st$ and $6th$ leak events with small leak sizes are successfully detected.

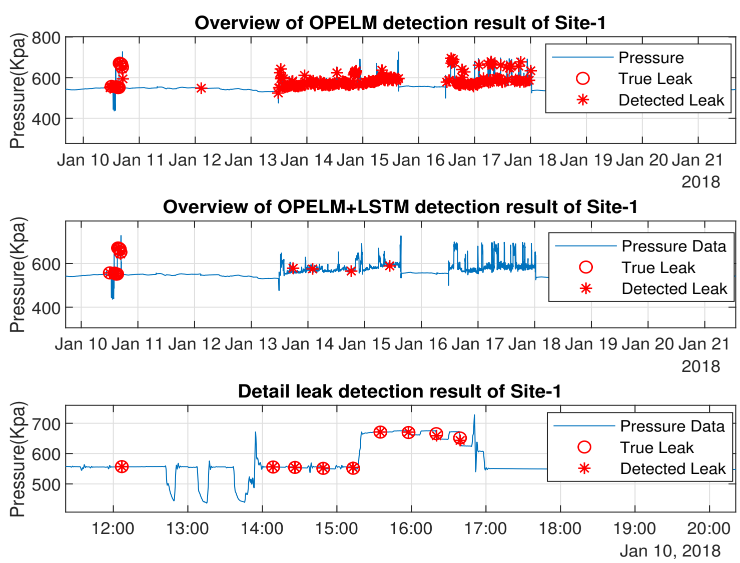

Figure 5.9: Detection result on Site-1.

2) Experiment on data from Site-2

The leak events description of Site-2 is listed in Table.5.4. There are 10 leaks occurred under different conditions in terms of flow rate and leak size.

Table 5.4: Experiment environment of Site-2.

Index	*Pipe Flow*	*Leak Flow*	*Leak Percentage*	*Detected*
$1-3$	$100m^3/day$	$20m^3/day$	20%	Yes
$4-7$	$20m^3/day$	$1m^3/day$	5%	No
$8-10$	$10m^3/day$	$0.25m^3/day$	2.5%	Yes
Data Length: 487 *hours* Leak Event: 10 Content:Salt water				

- First stage result of Site-2: The top plot in Fig. 5.10 shows the first-stage detection result of Site-2. It can be observed that similar to the result of Site-1, many alarms have been reported including the true and false ones. In this case, totally 138 alarms are detected where only 10 TP events exist. The suspicious leaks mainly appear when there is a sudden drop of the pressure wave. Although the FP number is less than that in Site-1, the false alarms appearing every few hours may still jeopardize the entire detection efficiency.

- Second stage result of Site-2: The middle and bottom plots in Fig. 5.10 show the final detection result of Site-2. The number of false alarms is significantly decreased from 138 to 13 in the 487 hours long data set. In the detailed view of the second-stage result, 2 leak events are missed. In fact, it should be noted that the second-stage BiLSTM is adopted to mainly remove the large number of false alarms that jeopardizing the efficiency of the leak detector. However it may induce a slight increase of missed leak detection. By a close inspection of the two missed leaks, it can be found that their dropping trends are not smoothly downward like other leak events. There exist dropping rate variations in the pressure wave. The proposed method mistakenly take these two events as noises instead of TP. In general, most of the leak events are accurately detected with very few false alarms in the final result. Even for the small leaks $8-10$ listed in Table. 5.4, they are successfully detected.

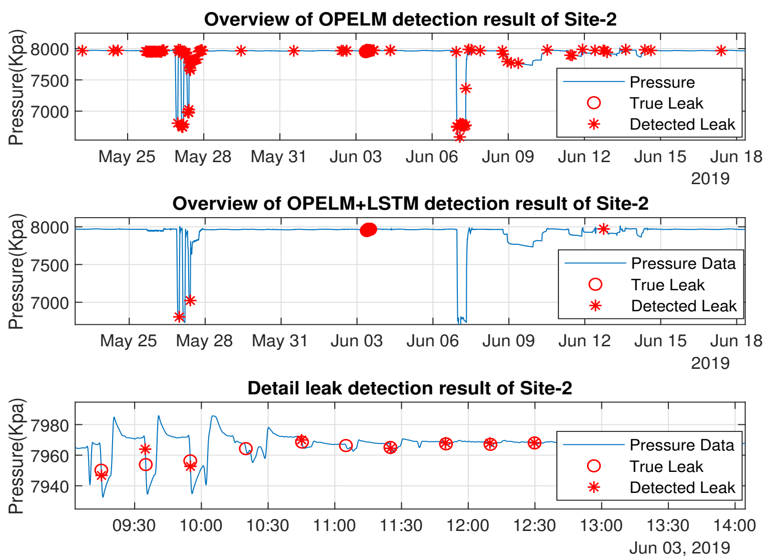

Figure 5.10: Detection result on Site-2.

3) Experiment on data from Site-3

Table 5.5 shows the leak status of Site-3 experiment. There are 18 leak events under multiple pressure and leak flow conditions. It can be observed from Fig. 5.11 that the pressure wave is fluctuating continuously in a large amplitude. The noisy pressure fluctuation is the normal working state caused by pumping or other pipeline operations. However, the sudden and continuous pressure change may lead to numerous FP and deteriorate the detection performance.

- First stage result of Site-3: The top plot in Fig. 5.11 shows the first-stage detection result. In this case, there are 460 suspicious leak events detected in the 668 hours long experiment data. Comparing to the working conditions of Site-1 and Site-2, the pressure wave fluctuates continuously which may increase the amount of false alarms.

Table 5.5: Experiment environment of Site-3.

Index	*Pipe Flow*	*Leak Flow*	*Leak Percentage*	*Detected*
$1-3$	$347.5m^3/day$	$55m^3/day$	16%	Yes
$4-6$	$347.5m^3/day$	$15.6m^3/day$	4.5%	Yes
$7-9$	$347.5m^3/day$	$8.7m^3/day$	2.5%	Yes
$10-12$	$82.5m^3/day$	$20.6m^3/day$	25%	Yes
$13-15$	$82.5m^3/day$	$3.7m^3/day$	4.5%	Yes
$16-18$	$82.5m^3/day$	$2.0m^3/day$	2.5%	No
Data Length: 668 *hours* Leak Event: 18 Content:Salt water				

- Second stage of Site-3: On the second stage as shown in the middle and bottom plot in Fig. 5.11, the number of alarms decreased from 460 to 25 where 15 alarms are TP. The last three leak events are missed. It can be seen from Table.5.5 that the last three leaks are relatively smaller than the others, at the same time the flow rate is also lower. It indicates that under the conditions of low pipeline flow rate and noisy pressure environment, small leaks may not be successfully detected.

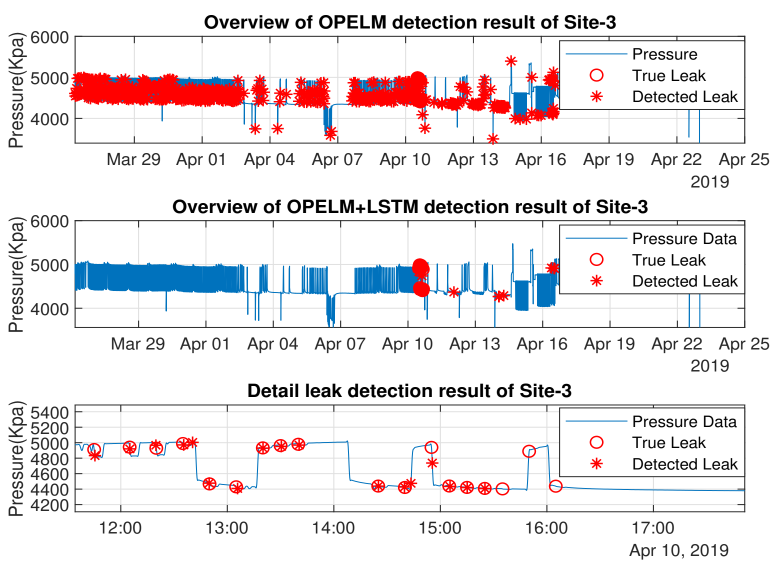

Figure 5.11: Detection result on Site-3.

4) Experiment on data from Site-4

Table 5.6 shows the leak status of Site-4 experiment. There are 19 leak events under various pressure and leak flow conditions in a noisy environment. The $8th$, $9th$ and $19th$ leak events are failed to be detected. The details are shown below.

Table 5.6: Experiment environment of Site-4.

Index	*Pipe Flow*	*Leak Flow*	*Leak Percentage*	*Detected*
$1-4$	$200m^3/day$	$28.4m^3/day$	14.4%	Yes
$5-7$	$200m^3/day$	$10m^3/day$	5%	Yes
$8-10$	$200m^3/day$	$5m^3/day$	2.5%	No
$11-13$	$150m^3/day$	$34m^3/day$	22.66%	Yes
$14-15$	$150m^3/day$	$7.5m^3/day$	5%	Yes
$16-17$	$150m^3/day$	$30m^3/day$	20%	Yes
$18-19$	$150m^3/day$	$7.5m^3/day$	5%	No
Data Length: 670 *hours* Leak Event: 19 Content:Salt water				

- First stage result of Site-4: As shown in the top plot in Fig. 5.12, in the first stage, 280 suspicious leaks are alarmed. Most of them appear at the edges of the dropping pressure wave. Obviously, the frequently appearing pressure drops are not leaks. By taking the advantage of its ambient pressure characteristics, false alarms can be discriminated in the next step.

- Second stage result of Site-4: In this stage, the number of alarms is reduced from 280 to 29. As shown in the middle and bottom plots of Fig. 5.12, the $8th$, $9th$ and the $19th$ leak events are not successfully detected. From Table.5.6, their leak sizes are relatively smaller than the others. Moreover, by viewing the pressure wave of the $8th$ leak, it exists in a rising pressure trend which is opposite to the typical dropping trend of a leak. It can be seen that when a leak occurs in the uprising part of a pressure wave, the leak may not be successfully detected.

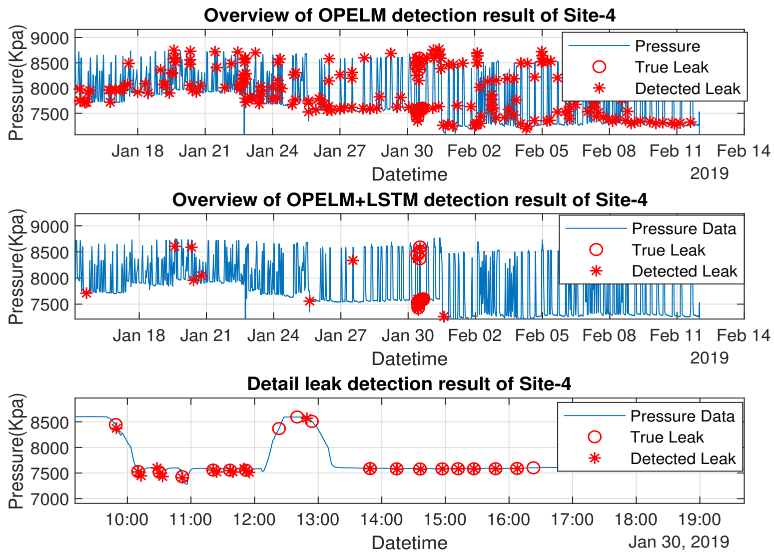

Figure 5.12: Detection result on Site-4.

5.3.3 Detection Performance Comparison

In this section, several conventional ML based leak detection methods such as the BPNN in [137], SVM in [138], KNN in [141], ELM in [142], NB in [144] and DT in [145] are compared with the proposed method, and BiLSTM is also tested in the first-stage detection. Furthermore, to verify the effectiveness of BiLSTM in eliminating false alarms, the aforementioned ML methods with BiLSTM added as the second stage are also implemented. The performances are evaluated on 100 rounds of experiments on data from Site-2.

1) Performance comparison between various ML methods and the proposed method

In this section, two key factors including the number of false alarms (FP) and the number of false detection (false alarm and missed alarm) are adopted for comparison among various ML methods. The ratio between results from other investigated methods and the proposed one are shown in Fig. 5.13.

It can be observed that the proposed method achieves the least number of false alarms and false detection. The second best result is achieved by

ELM. Among all other ML methods, Naive Bayes method has the worst performance where the amount of false alarms and false detection are almost 1000 times greater than the proposed method. Notably, the performance of applying BiLSTM only without the first stage detection is not satisfactory, it may be even worse than other conventional methods such as SVM, KNN and ELM. Although the strong memorizing ability of BiLSTM can be utilized to effectively identify long-short term sequence patterns, its sensitivity to the transient change may not be comparable to other methods so that results in worse detection performance.

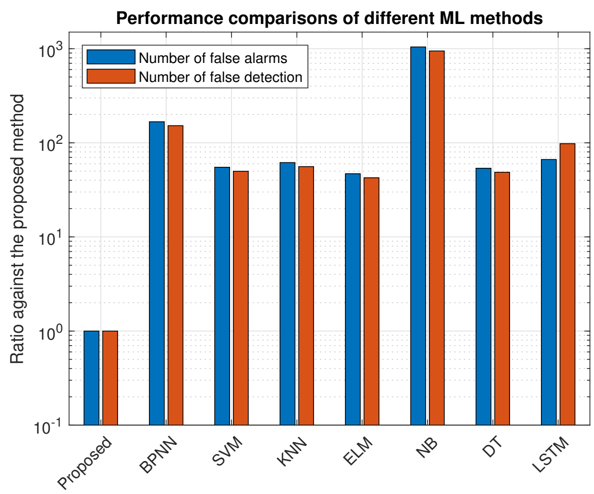

Figure 5.13: Performance comparison among different ML methods.

2) Performance comparison between various ML methods after attaching BiLSTM

Fig. 5.14 shows the performance comparison between the proposed method and various ML methods combined with the BiLSTM to improve the detection performance.

It can be seen that all the investigated methods achieve much less false alarms and false detection when attaching BiLSTM as the second stage to identify false alarms. Take the SVM+LSTM as an example, the false alarm number decreases from 190 to 4 and the false detection number drops from 4800 to 130 out of 116160 samples. It verifies that by considering a broader temporal range of observations around the suspicious leaks, the false alarms can be effectively eliminated.

Among all the ML methods combined with BiLSTM, the proposed method still achieves the best performance, owing to the excellent classification performance of the OPELM.

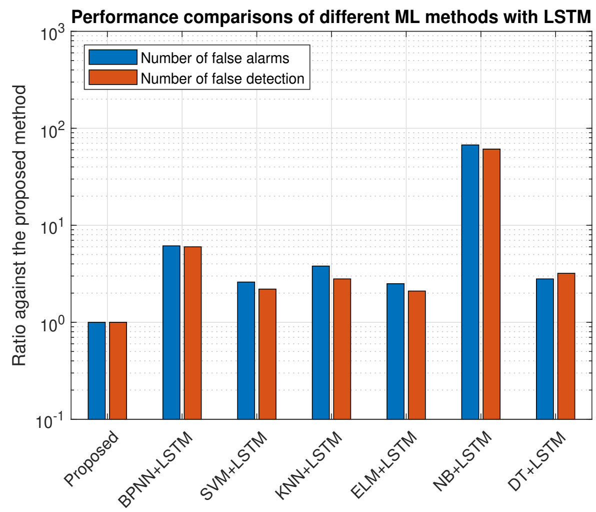

Figure 5.14: Performance of improved ML detection methods.

5.3.4 First-stage Classifier Selection

To choose the most appropriate classifier to perform the first stage detection, some comparisons in terms of learning speed and test accuracy are included. The learning speed and accuracy among aforementioned methods are shown in Table. 5.7. The experiment is based on the training set sized at 20000 within which the amounts of class0 and class1 samples are equal. The numbers shown are the averaged value of 100 experiments. For methods such as BPNN, ELM and OPELM, the number of hidden layer neurons are set to be 40. The kernels in SVM and OPELM are chosen as radial basis function (RBF).

It can be seen that KNN and NB have the fast learning speed, but the test accuracy is obviously lower than OPELM and SVM. The three classifiers with test accuracy higher than 0.9 are SVM, ELM and OPELM, however, SVM takes much longer in learning. The accuracy of ELM is obviously lower than OPELM although the learning speed is faster. Therefore, considering both of the learning speed and test accuracy, OPELM is deemed the most effective one among the investigated methods to perform the first-stage leak detection in this study.

Table 5.7: Learning speed and accuracy comparison.

Methods	*BPNN*	*SVM*	*KNN*	*NB*	*ELM*	*OPELM*
Time(s)	9.54	3.26	0.06	0.02	0.16	0.81
Accuracy	0.854	0.972	0.863	0.832	0.922	0.974

5.3.5 Effects of Template Length

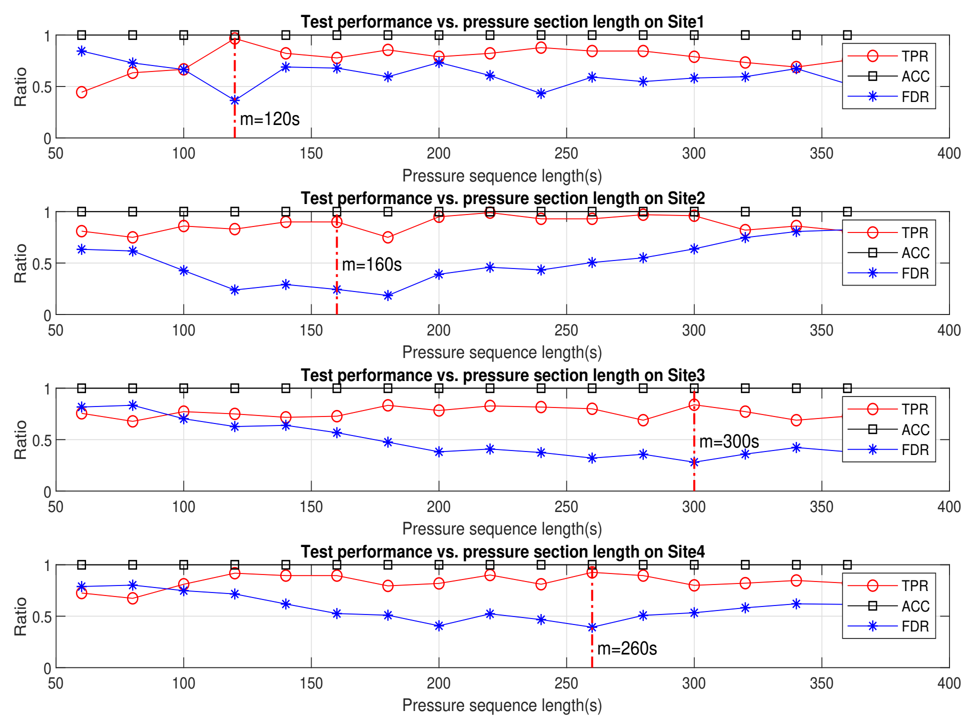

Figure 5.15: Detection performance vs. pressure section length m.

The length of pressure template m defines the time span of the pressure portion used for feature extraction. The value of m is closely related to the quality of feature representation. When m is too small, the template may not be able to represent the entire leak pressure characteristics, and too large m may lead to adverse influence by non-leak pressure wave, thus deteriorates the feature representation. The appropriate value of m is dependent on multiple factors affecting waveform fluctuations such as pipe flow rate, leak size, pipe content, pump or valve operations, etc. It usually varies from case to case.

Fig. 5.15 demonstrates the detection performance with respect to different template length m of the four investigated data sets. All ACC results of the 4 sites indicating the overall detection accuracy are close to 1, which show that ACC is not sensitive to m since the detection error is generally small.

Hence, to choose a proper value of m, both of TPR and FDR are considered. Higher TPR means higher percentage of leak events are to be successfully detected. However, FDR is expected to be small because it indicates the percentage of false alarms counted in all alarms. Therefore, the value of m is chosen at which the difference between TPR and FDR is maximum. As can be observed in Fig. 5.15, the first two m values are 120s and 160s respectively. When the intensity of fluctuations increases, it will need longer templates to achieve a satisfied detection performance. The fluctuations in Site-3 and Site-4 are more dramatic than that in the first two cases, therefore m is chosen as 300s and 260s respectively.

5.4 Conclusion

In this chapter, a novel PPA leak detection method is proposed based on supervised OPELM combining BiLSTM for continuous pressure monitoring leak detection system. The contributions of this chapter are summarized as follows.

Firstly, the proposed method can achieve higher detection accuracy with significantly lower false alarm rate than existing ML based PPA leak detection methods.

Secondly, the advantage of strong memorizing capability of BiLSTM is firstly utilized to discriminate the true and false alarms. The effectiveness is verified by experiments on different real-world data sets.

Finally, the proposed feature extraction scheme can effectively represent the characteristics of leak pressure transient, thus, enhance the detection performance. Furthermore, a thorough comparison study is performed.

However, when minor leaks occur resulting in very subtle pressure variations, or when leaks coincide with abrupt up-trend (i.e., increasing) pressure changes, they may not be effectively detected by the currently proposed approach. In this case, other detection methods with higher sensitivity can be investigated, but the trade-off between sensitivity to minor leaks and false alarm rate needs to be carefully considered.

Chapter 6

Pipeline Leak Detection and Disturbance Assisted Localization Based on BiLSTM*

6.1 Introduction

Pipeline transportation has become one of the most important measures to convey fluid industrial products such as oil, gas, water, etc. However, the pipeline leakage can cause huge property losses and environment hazards especially when the transported product is flammable, poisonous or corrosive. To ensure the safety of pipeline networks, leak detection and localization methods have been comprehensively studied. Continuous pressure monitoring is one of the most straightforward methods which inspects the pipeline pressure status acquired by sensors mounted along the pipeline. When a rupture or burst occurs, an abrupt negative pressure wave (NPW) propagates towards each side from the leak spot. By capturing the sudden pressure drop, a leak event can be detected. As an important application of WSNs, multiple pressure sensors are usually deployed, thus, by analyzing the time difference of arrival (TDOA) of NPW, given the NPW speed and pipeline length, the leak location can be obtained [152, 153].

To effectively detect leak events based on collected pressure data, recently,

*A version of this chapter has been submitted for publication as "Pipeline leak detection and disturbance assisted localization method based on BiLSTM," in *IEEE Sensors Journal*.

various ML methods have been proposed for pressure monitoring based leak detection systems. However, the issue of high false alarm rate is challenging. For example, the frequently occurred abrupt negative pressure transients under normal conditions are possibly mis-classified as leaks. On the other hand, when a small leak occurs, showing mild or weak leak characteristics, it may not be captured in time until the pressure finally drops below the alarm threshold, leading to more severe consequences. It is shown that if the temporal correlation of pressure data is taken into consideration, the aforementioned challenges can be effectively tackled. DL based RNN provides a promising tool to perform classification considering the temporal correlation. In [154], LSTM neural networks are applied to address the challenge of temporal correlation and the detection performance is verified through experiments. In [155], BiLSTM is utilized to further identify the leak detection results from ELM networks. It is shown that the false alarms can be dramatically reduced by considering the temporal correlation in an expanded time range. However, both [154] and [155] are only focused on leak trend recognition, but the other non-leak pressure disturbances (or transients) are simply ignored. In fact, if the non-leak pressure disturbances can also be identified, they may provide useful information such as NPW speed to improve the accuracy of leak localization.

To locate a leak spot, methods based on time difference of arrivals (TDOA) of NPW are commonly used. In such a method, the accuracy of localization depends largely on the accuracy of calculated time lags and the NPW propagation speed adopted. In [156], instead of using a predefined value for the NPW propagation speed, a leak localization method is proposed which estimates the speed by analyzing TDOA of two proximate sensors. By exploiting the property of amplitude attenuation, the leak localization accuracy is improved. However, in this work the verification experiment is implemented on a $200m$ long pipeline, which may be insufficient to represent the practical long distance pipeline scenarios. In [157], a weighted average localization algorithm based on TDOA is proposed and the experiment results show improvement comparing to conventional methods. Furthermore, in [158, 159], the time-frequency analysis and data fusion techniques are applied in fluid pipeline leak localization, rendering improved performance, especially in detecting multiple small leaks. In addition, in [160] the wavelet transform (WT) based leak localization methods are proposed and experimented where the TDOA is retrieved by wavelet analysis but the NPW speed is assumed as constant. In [161], the WT based TDOA estimation method is compared with traditional minimum-searching and cross-correlation based methods, and it is shown to outperform them in most scenarios. But in spite of that, the performance of WT method is still largely case dependent and may not be readily generalized. It should be noted that in these works the speed utilized in localization is assumed constant which limits its accuracy in many practical situations.

To recapitulate, in the existing TDOA based leak localization methods, the NPW propagation speed is either assigned an empirical value, or calculated from leak impacted pressure waves and assumed as constant afterwards. The information existed in most common appeared disturbances are usually ignored, which can be exploited for NPW speed estimation and online update, thus, the improved leak localization accuracy can be obtained. In this chapter, a BiLSTM based pipeline leak detection and disturbance assisted localization method is proposed in multi-sensor monitoring scenarios. Main contributions of this chapter are elaborated as follows:

- A pressure sequence classification scheme via BiLSTM is developed to accurately recognize leaks and non-leak disturbances by incorporating temporal correlations in pressure transients.
- The identified non-leak disturbances are exploited to estimate and update the NPW propagation speed.
- The proposed method is validated on both simulated and real-world pipeline leak experiment data sets, through comparison case studies.

The results demonstrate that it can be used to reliably detect leak events and improve the localization accuracy compared to conventional methods.

The remainder of this chapter is organized as follows. Section 6.2 introduces the motivation and problem statement. Section 6.3 presents the details of the proposed method. Section 6.4 includes the experiment and comparison results with discussions. Finally, conclusion is drawn in Section 6.5.

6.2 Problem Statement

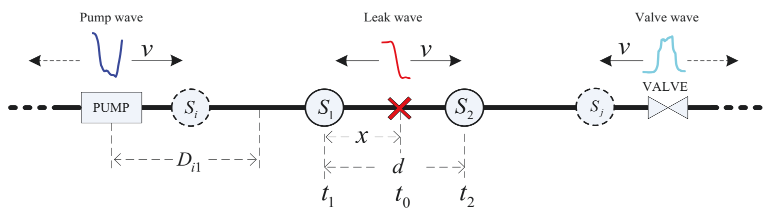

Figure 6.1: Illustration of an example scenario.

An example scenario of NPW based leak detection and TDOA based localization is shown in Fig.6.1. In conventional methods, e.g. [157], when a leak or rupture occurs between pressure sensors S_1 and S_2 at time t_0, the NPW caused by the leak propagates at speed v towards two opposite directions. The pressure inflection measured by S_1 and S_2 triggers the detection system to flag a leak and its location can be calculated by

$$x = \frac{d + v \cdot \Delta t}{2} \tag{6.1}$$

where x is the distance between leak spot and S_1, d the distance between S_1 and S_2, $\Delta t = t_1 - t_2$ is the time difference of NPW arrival at S_1 and S_2. In addition, as shown in Fig. 6.1, D_{i1} represents the distance between a pump and the center of sensor pair $S_{i1} = \{S_i, S_1\}$.

To enhance localization performance, in most existing methods, more efforts are made to improve accuracy in calculating TDOA, Δt, while a predetermined constant NPW speed v is used, [160, 161]. However, in reality v can be varying over time which may lead to inaccurate localization result. Moreover, numerous non-leak pressure disturbances caused by pump or valve operations commonly exist, which are prone to be misidentified as leaks, causing excessive false alarms and compromising the reliability of leak detection.

In view of these drawbacks, we propose to identify not only the leak trends but also the recurrent non-leak disturbances by utilizing the strong sequence classification ability of BiLSTM. Then, the TDOA of non-leak disturbances is calculated online to update the NPW speed v, which is subsequently used in localizing a leak upon its detection. This way, not only the leaks can be detected but also the non-leak disturbances can be utilized to obtain the most updated NPW speed such that the localization is performed with improved accuracy.

6.3 Proposed Method

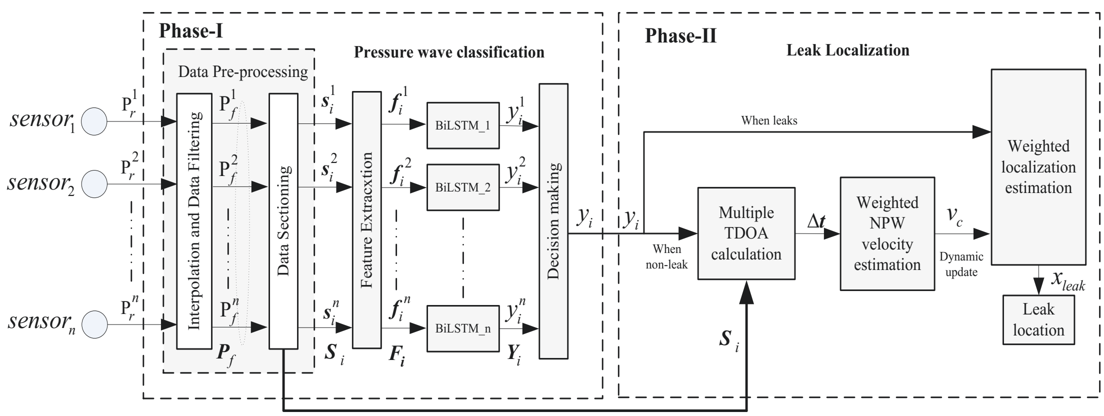

Figure 6.2: Block diagram of leak detection and localization process.

The diagram of proposed method is shown in Fig. 6.2. In Phase-I, pressure waves from different sensors are collected, pre-processed and classified into "Leak" or "non-leak" waves (and the latter is referred to as disturbances). In Phase-II, the NPW speed is online estimated by calculating the time lags of pressure disturbances from different sensors when "non-leak" is reported by Phase-I. On the other hand, when "Leak" is reported, leak localization is performed by using TDOA method with the newly update NPW speed.

In the following, each step of the proposed scheme is presented with details.

6.3.1 Pressure Wave Classification Phase

1) Data pre-processing

In practice, sensors installed along the pipeline may have different sampling rates and contaminated with noises which may induce errors in the following TDOA calculation. To ensure the synchronization and remove the high frequency noises, raw pressure sequences from n sensors $\boldsymbol{P}_r = [\mathrm{P}_r^1, \cdots, \mathrm{P}_r^n]$ are implemented with interpolation and low-pass filtering. In this chapter, the sampling period after interpolation is $0.1s$ and the maximal overlap discrete wavelet transform (MODWT) [162] is utilized to denoise the signal with Donoho and Johnstone's universal threshold and level-dependent thresholding, where the wavelet function is selected as "$db10$" and the level of wavelet transform is 4. The filtered data $\boldsymbol{P}_f$ is then sectioned and prepared for feature extraction.

2) Features extraction

To enhance the feature representation and improve the classification performance, a similarity based feature extraction scheme is proposed. The process of extracting features from sensor j is shown in Fig. 6.3.

A sliding window is utilized to select a section of pressure wave and calculate the similarity to typical signatures. The calculated similarities are taken as features. For the example shown in Fig. 6.3, three features are obtained as "Similarity to leak", "Similarity to pump" and "Similarity to valve" respectively. Denote the features extracted from the i-th section $\boldsymbol{s}_i^j$ of sensor j as $\boldsymbol{f}_i = [f_{i1}, f_{i2}, f_{i3}]$, the feature sequences of this sensor can be written as $\boldsymbol{F} = \left[\boldsymbol{f}_1^T, \cdots, \boldsymbol{f}_i^T, \cdots, \boldsymbol{f}_m^T\right]^T$.

- Typical signatures characterization: To choose the most representative signature of the corresponding class, the pressure section that is most

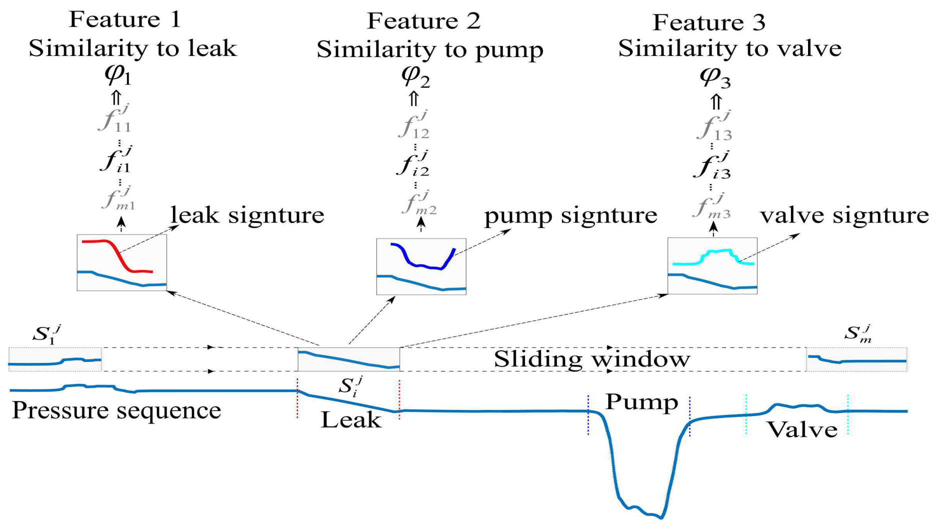

Figure 6.3: Diagram of feature extraction.

similar to the others in the same class but dissimilar to those of different classes should be chosen. The scheme of choosing the typical signature was introduced in Chapter 5.2.4, in this case, three types of signatures are to be chosen which are "Leak", "Pump" and "Valve".

- Similarity calculation via dynamic time warping (DTW): The conventional inner product based similarity calculation works well when the two sequences are in the same length, however, the lengths of typical signatures and the compared pressure sections are rarely the same. Therefore, dynamic time warping algorithm is employed. Fig. 6.4 shows an example of warping a pump signature and a pressure section with different length [163]. Compared to the original signals on the left, the warped signals on the right are properly matched. The similarity calculation based on the warped signals is defined as the exponential function of DTW distance in Eqn. (6.2).

$$f_i = e^{-\frac{DTW(P_i,P_s)}{2}}, \tag{6.2}$$

where f_i is the similarity between the i-th pressure section P_i and its typical signature P_s.

By applying DTW algorithm, the calculated similarity feature sequences $\boldsymbol{\varphi}_1$,$\boldsymbol{\varphi}_2$ and $\boldsymbol{\varphi}_3$ are shown in Fig. 6.5. The light green parts in each plot indicates the matched typical pressure sections, also plotted are sequences of similarity scores. It shows that the similarity in each of the three plots achieves relatively high value when the corresponding typical section appears. Therefore, it can effectively indicate the presence of a typical section with respect to each signature, namely, leak, pump, or valve related.

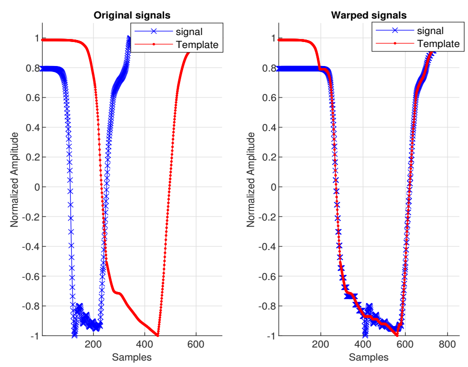

Figure 6.4: Example of similarity calculation by DTW.

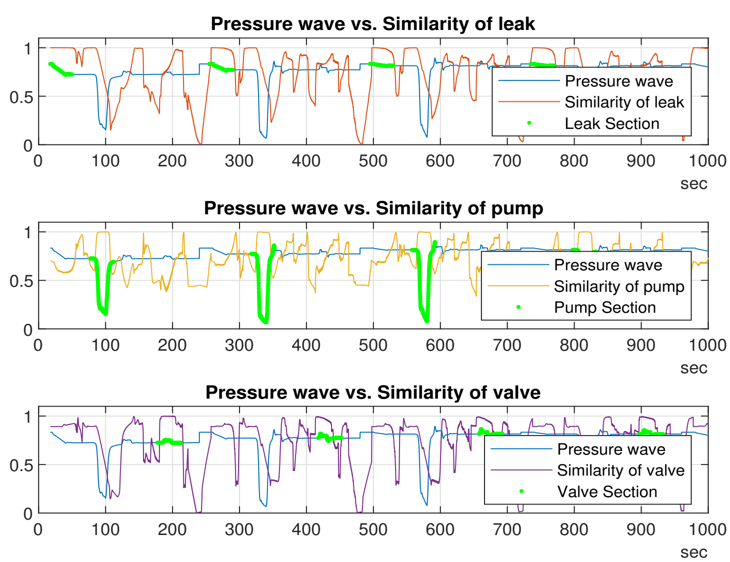

Figure 6.5: Example of feature sequences sectioning.

To use BiLSTM for classification, in this chapter, the training set is chosen from the labeled feature sequences $\boldsymbol{F}$ and the corresponding labels Y as follows.

$$\boldsymbol{F} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ \vdots & \cdots & \vdots \\ f_{i1} & f_{i2} & f_{i3} \\ \vdots & \cdots & \vdots \\ f_{m1} & f_{m2} & f_{m3} \end{bmatrix} \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_m \end{bmatrix} \tag{6.3}$$

The element y_i in Y indicates the class label that the corresponding feature belongs to, e.g., $y = C \in \{0, 1, 2, 3\}$, referring to $\{Leak, Pump, Valve, Normal\}$.

To select the training set from feature matrix $\boldsymbol{F}$ and label vector Y, for example a training set for leak event, the following procedure is applied. Suppose the i-th sliding window overlaps with a known leak section, the selected training set is written as:

$$\boldsymbol{F}_i = \begin{bmatrix} f_{i-\tau,1} & f_{i-\tau,2} & f_{i-\tau,3} \\ \vdots & \cdots & \vdots \\ f_{i,1} & f_{i,2} & f_{i,3} \\ \vdots & \cdots & \vdots \\ f_{i+\tau,1} & f_{i+\tau,2} & f_{i+\tau,3} \end{bmatrix} \quad y_i = 0, \tag{6.4}$$

where $\boldsymbol{F}_i$ are the selected feature sequences centered at instant i and spanned τ instants on each side. y_i is the $Leak$ label. The value of τ determines the length of sequences used for training the classifier. When $\tau = 50$, there are 101 points in a sequence, suppose the sliding window step $\delta = 5s$ and window size $w = 35s$, the time duration covered by selected feature sequences is calculated as $T = 2\tau \times \delta + w = 535s$. Therefore, the pressure status within the $535s$ are employed to train the BiLSTM classifier for leaks.

3) Results pooling and decision making

In multiple sensors scenario shown in Fig. 6.2, each classifier produce its own result. To make the final decision based on multiple results, in this chapter, a voting mechanism is adopted. The class with the maximum number of votes is selected as the final decision. Using the voting mechanism, individual false result may be corrected and the classification accuracy is improved.

6.3.2 Leak Localization Phase

The leak spot localization is based on the TDOA method as introduced in Eqn. (6.1). The knowledge needed in TDOA method are the sensor distributions, the time difference of arrival Δt and the NPW speed v. The main process of propose leak localization method is given in Alg. 6 at first, followed by detailed illustrations of each key step.

Algorithm 6 Main steps of the proposed localization process.

1: Obtain classification result Y_i from Phase-I when classifying the i-th pressure section as shown in Fig. 6.3.
2: Calculate the lags of the arrival time $\boldsymbol{\Delta t}$ between each pair of the n pressure sensors.
3: If Y_i indicates one of the non-leak classes, first calculate the NPW propagation speed v_{ij} of each sensor pair based on $\boldsymbol{\Delta t}$, where i and j represent two different sensors; then obtain the weighted sum v_{ij} to update the current speed v_c;
4: If Y_i indicates leak, apply weighted TDOA method to calculate the leak location by Eqn.(6.1) and Eqn.(6.7) employing the updated NPW speed v_c.

1) Calculation of TDOA

To obtain the time difference of arrival Δt, various of signal processing techniques can be utilized and each of them owns its unique advantage and disadvantage [161]. In this chapter, we adopt the cross-correlation based method due to its simplicity and low complexity. Because the pressure wave is usually distorted when arriving at different sensors. Directly calculating the cross-correlation of two distorted waves may not accurately render the time lags. Therefore, in this chapter, instead of calculating the cross-correlation of two pressure sequence directly, we firstly calculate the cross-correlation of the pressure wave and its own signature, then compare the two coefficient sequences to find time lags.

Fig. 6.6 shows an example of calculating the Δt of sensor 1 and sensor 2 when classification result is $Valve$. The pressure sequence and the corresponding signatures are displayed in the top and middle plots of Fig. 6.6, the cross-correlation coefficients are displayed in the bottom plot. By comparing the time lag of the maximum coefficient values, the time difference is obtained. In this case, the time difference is $4.75s$.

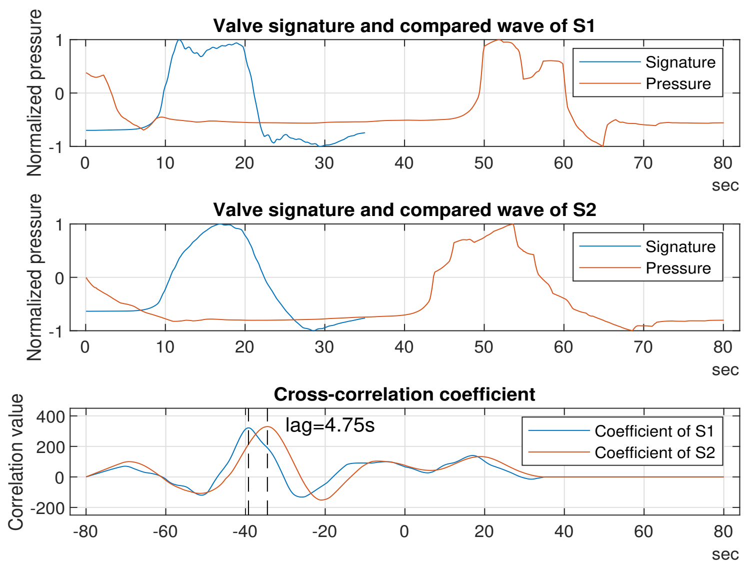

Figure 6.6: Example of time lag calculation.

2) Weighted NPW Velocity

When non-leak disturbance is detected, in the case of multiple sensors of Fig. 6.1, multiple NPW speeds v_{ij} can be obtained by the TDOA of sensor i and sensor j. A weighted calculation is adopted, as shown in Eqn. (6.5) and Eqn. (6.6).

$$v_c = \frac{1}{M} \sum_{i=1,\cdots,n-1}^{j=i+1,\cdots,n} \frac{1}{d_{ij} \cdot D_{ij}} \cdot v_{ij} \tag{6.5}$$

where

$$M = \sum_{i=1,\cdots,n-1}^{j=i+1,\cdots,n} \frac{1}{d_{ij} \cdot D_{ij}} \tag{6.6}$$

It considers not only the distance d_{ij} between sensors i and j, but also the distance between the excitation source and a sensor pair, D_{ij}, which can both inversely influence the estimation accuracy. In calculating the weighted speed v_c, more weights are assigned to the value v_{ij} produced by the sensor pair with shorter distance and are closer to the excitation source.

3) Weighted Leak Localization

After obtaining the NPW speed and TDOA between two sensors, the leak location can be calculated by Eqn. (6.1). However, in the multi-sensor scenario as shown in Fig. 6.1, multiple TDOA $\Delta\boldsymbol{t} = [\Delta t_{i2}, \Delta t_{ij}, \Delta t_{12}, \Delta t_{1j}]$ can be used to calculate leak location. A weighting scheme as introduced in [157] is applied to get the final leak location. The scheme is described as follows:

$$x_{leak} = \sum_{i=1}^{N_p} \alpha_i x_i \tag{6.7}$$

$$\alpha_i = \frac{\frac{1}{d_i}}{\sum_{i=1}^{N_p} \frac{1}{d_i}} \qquad s.t. \quad \sum_{i=1}^{N_p} \alpha_i = 1,$$

where N_p is the number of sensor pairs for which their distances and TDOA are used to calculate the leak locations. x_i is the leak location calculated by the i-th pair. α_i is the weight assigned to the leak location result x_i. d_i is the distance between the two sensor nodes of the i-th pair. x_{leak} is the weighted leak location result.

6.4 Simulation and Experiment

In this section, both simulation and experiment case studies are conducted to examine the proposed method. The first case study is performed on the scenario simulated by the software tool "ALLIEVI". The second case study is performed on real-world experimental data acquired from a crude oil pipeline. In both cases, two layers of hidden units are employed in the structure of BiLSTM where the unit numbers are 125 and 100 respectively. The optimal number of hidden units can be case dependent which is not discussed in this chapter.

6.4.1 Case-I: Simulated Pipeline Leak Scenario

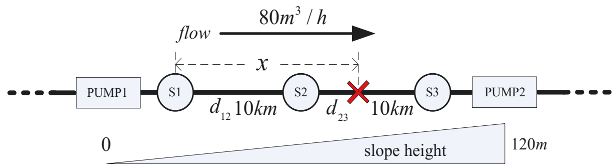

Figure 6.7: Case-I simulation scenario.

The scenario simulated in Case-I is shown in Fig. 6.7. The length of pipeline is $20km$ and sits between two pump stations. There are 3 uniformly distributed pressure sensors mounted along the pipeline. The elevation is rising $120m$ with constant slope rate. The NPW speed is set to follow a slow varying sinusoidal between $1000m/s$ and $1400m/s$ in different simulations and it is assumed constant in each simulation. The simulated pressure head is between $10m$ to $300m$. There are 20 leak events occurring at random locations between sensor S_1 and sensor S_3, the distance from the leak spot to S_1 is denoted as x. More details on the parameters setting can be found in Tab. 6.1.

Table 6.1: Environment parameters of the simulated pipeline system.

Pipe Model	*DN200*	*Diameter*	$202.74mm$	*Thickness*	$8.18mm$
Sensors	*3*	*Length*	$20km$	*Flow*	$19.6L/s$
Ave. speed	$1200m/s$	*Min. Leak*	$10m^3/h$	*Max. Leak*	$60m^3/h$
Length d_{12}	$10km$	*Length* d_{23}	$10km$	*Elevation*	$120m$
Leak Events	20	*Pump1 Events*	200	*Pump2 Events*	100
Total Points	6.4×10^5	*Sampling Rate*	$0.1s$	*Sequence Length*	$35s$

1) Leak detection performance

In the simulation, 10^5 time instants are randomly selected. Centered with each selected time instant, the pressure sequence within $35s$ is performed with sequence classification by BiLSTM classifier. The classification results are marked in corresponding color at the selected center point. Fig. 6.8 gives a section of the sequence classification result where the identified “Pump1”,“Leak”, “Pump2” and “Normal” are marked in black, red, blue and green respectively. It can be observed in the red circles that there exist several false alarms. However, the final classification result is based on the voting mechanism where *Sensor* 1 and *Sensor* 3 are not reporting leak at the same instants, so the final classification result is not affected by the false alarm of *Sensor* 2. In this case study, the events regarding “Pump1”, “Pump2” and “Leak” are all correctly detected with no false or miss alarm in the final result.

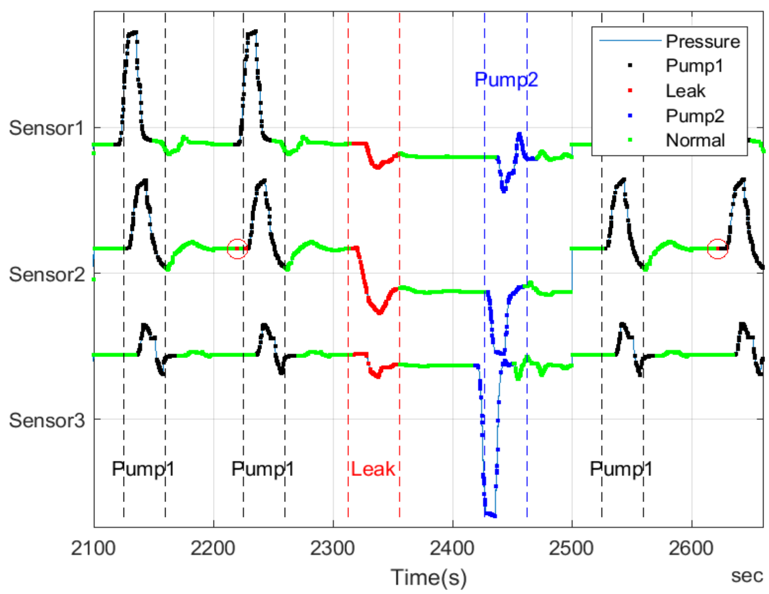

Figure 6.8: Sequence classification example of Case-I.

To further investigate the classification performance of each classifier corresponding to each sensor, i.e., S_1, S_2 and S_3, the detailed statistics of classification results are provided in Tab.6.2. Some common used measures are employed for statistics such as TPR, FDR and Accuracy as introduced in Chapter 5.3.1. Simulation results verify that the BiLSTM based classifier can effectively identify different pressure sequences with higher than 90% overall classification accuracy. In [155], the effectiveness and performance of BiLSTM

Table 6.2: Classification result statistics of Case-I.

Label		*Classifier 1*	*Classifier 2*	*Classifier 3*
Leak	*Num.*	2451	2502	2529
	TPR	98.94%	97.44%	97.07%
	FDR	1.97%	5.18%	0.64%
Pump1	*Num.*	21967	21802	22077
	TPR	98.92%	96.65%	98.50%
	FDR	0.01%	1.26%	0.8%
Pump2	*Num.*	10925	10983	11169
	TPR	96.92%	97.01%	98.64%
	FDR	5.16%	3.23%	0.08%
Normal	*Num.*	64657	64713	64225
	TPR	89.91%	86.57%	97.96%
	FDR	1.7%	2.56%	1.56%
Accuracy		92.87%	90.19%	98.13%

based classifier for leak detection is discussed and compared with several other ML algorithms. It has been shown to outperform the other ML methods in leak detection accuracy.

2) Leak localization performance

In this experiment, the localization results by the proposed disturbance assisted method and conventional constant NPW speed method [157] are compared in Tab. 6.3. In conventional method, the fixed NPW speed is set to be $1200m/s$ which is the average of simulated speed values. There are 20 leaks at different locations under various NPW speed. The real NPW speed and the estimated values are listed in the $2nd$ and $3rd$ columns, it shows that the proposed method can effectively track the varying speed with average absolute error of $2.9m/s$. Also, the localization result of proposed method and conventional method are listed in the right part of Tab. 6.3. The localization error by proposed method is averaged at $24.8m$ which is much less than $307.2m$ by conventional method. It verifies that the proposed method can significantly improve the leak localization accuracy compared to traditional fixed NPW speed method.

Table 6.3: NPW speed and leak localization result of Case-I.

Index	*NPW Speed(m/s)*			*Location(m)*			*Error(m)*	
	True	**Estimated**	Err.	*True*	**Proposed**	**Con.**	**Proposed**	**Con.**
1	1228.2	1227.0	1.2	10649	10714	10800	65	151
2	1288.0	1290.3	2.3	9718	9722	9570	4	148
3	1339.2	1351.4	12.2	19772	19875	19350	103	422
4	1376.7	1379.3	2.6	14704	14690	14730	14	26
5	1397.0	1398.6	1.6	4758	4774	5160	16	402
6	1398.0	1398.6	0.6	11958	11960	12390	2	432
7	1379.6	1379.3	0.3	9951	9949	9630	2	321
8	1343.6	1342.3	1.3	3847	3786	4185	61	338
9	1293.6	1290.3	3.3	3530	3539	3825	9	295
10	1234.4	1234.6	0.2	12932	12898	12960	34	28
11	1171.8	1169.6	2.2	7450	7427	7425	23	25
12	1112.0	1117.3	5.3	19678	19717	20070	39	392
13	1060.8	1058.2	2.6	4251	4227	3795	24	456
14	1023.2	1030.9	7.7	17255	17237	17610	18	355
15	1003.0	1005.0	2.0	12077	12059	11490	18	587
16	1002.0	995.0	7.0	3047	3061	2160	14	887
17	1020.4	1020.4	0	4396	4391	3840	5	556
18	1056.4	1058.2	1.8	7072	7091	7035	19	37
19	1106.4	1105.0	1.4	12608	12600	12390	8	218
20	1165.6	1162.8	2.8	17632	17615	17700	17	68
Average absolute error			**2.9**	*Average absolute error*			**24.8**	**307.2**

6.4.2 Case-II: Real Industrial Experimental Data

Case study II is performed on the real-world data acquired from industrial site during leak experiment. The layout of experimented pipeline is shown in Fig. 6.9. There are three pressure sensors S_1, S_2 and S_3 deployed along the $22.98km$ pipeline, where the pipeline length between S_1 and S_2 is $14.79km$ and that between S_2 and S_3 is $8.19km$. The pipe content is mixture of crude oil and natural gas. The pressure range of the three monitoring points S_1, S_2 and S_3 are $20-400Kpa$, $300-700Kpa$ and $100-500Kpa$ respectively.

The sampling period is $1s$ originally and upsampled to $0.1s$ by interpolation. Totally 10 leak events are experimented under two different flow rates at the same location of $5km$ downstream of S_1. The detailed leak time and size information can be found in Tab. 6.4.

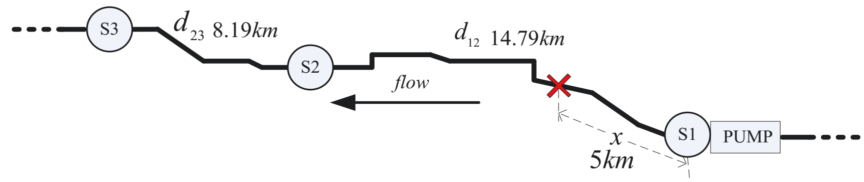

Figure 6.9: Leak experiment scenario of Case-II.

Table 6.4: Experimented leak events of Case-II.

Index	*Leak Spot*	*Pipe Flow*	*Leak Flow*	*Leak Size*	*Time Start*	*Time End*
1	$5000m$	$4.12m^3/h$	$0.06m^3/h$	1.45%	12 : 55	13 : 05
2	$5000m$	$4.15m^3/h$	$0.08m^3/h$	1.9%	13 : 30	13 : 40
3	$5000m$	$4.15m^3/h$	$0.19m^3/h$	4.6%	14 : 09	14 : 19
4	$5000m$	$4.16m^3/h$	$0.426m^3/h$	10.2%	14 : 26	14 : 36
5	$5000m$	$4.15m^3/h$	$0.835m^3/h$	20.1%	14 : 49	14 : 59
6	$5000m$	$4.15m^3/h$	$0.835m^3/h$	20.1%	15 : 13	15 : 17
7	$5000m$	$12.56m^3/h$	$0.125m^3/h$	1%	15 : 35	15 : 45
8	$5000m$	$12.75m^3/h$	$0.636m^3/h$	4.98%	15 : 57	16 : 07
9	$5000m$	$12.92m^3/h$	$1.296m^3/h$	10%	16 : 20	16 : 30
10	$5000m$	$12.52m^3/h$	$2.49m^3/h$	19.8%	16 : 39	16 : 49

It can be seen that the first 6 leak events are experimented under a lower pipe flow rate at about $100m^3/day$ with different leak sizes and the latter 4 events are experimented at a higher flow rate around $300m^3/day$. The increase of flow rate is at 15 : 18 by tuning up the pump output pressure at S_1 which may accordingly affect the NPW speed [164]. However, in this experiment the real NPW speed is absent, only the estimated values by proposed method are listed in Tab. 6.6 where the estimated NPW speed after 15 : 18 can be clearly observed increased and it verifies the inference of NPW speed varying.

1) Leak detection performance

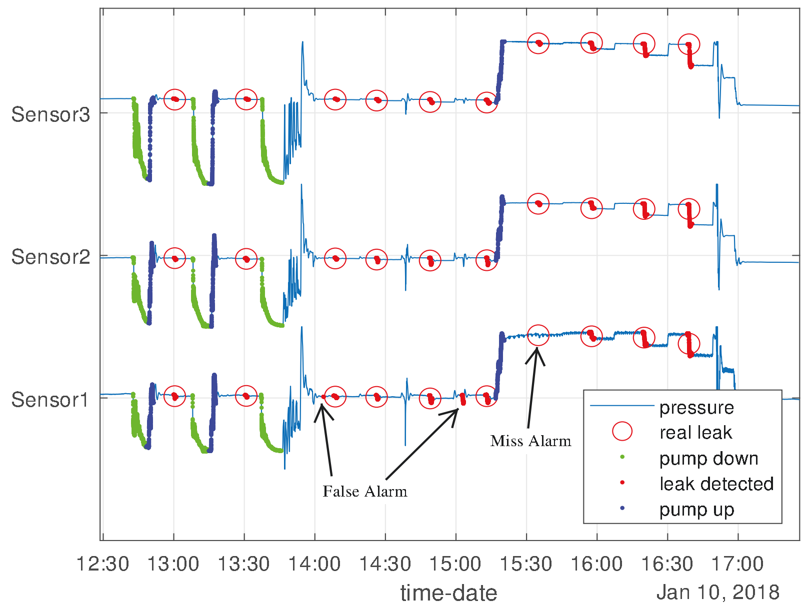

Figure 6.10: Sequence classification result of Case-II.

In this experiment, the pressure sequence length of $60s$ is sectioned for classification. The result is displayed in Fig. 6.10 where the detection result for "Normal" is hidden for clearer view. It shows that all the 10 leak events are successfully detected and the "Pump down" and "Pump up" patterns are also detected which can be utilized to online update the NPW speed. Some false alarms appear at $14:02$ and $15:03$ of *Sensor* 1 and a miss alarm is occurred at $15:35$. However, the correct results are provided by *Sensor* 2 and *Sensor* 3 at those time instants. The wrong detection results of *Sensor* 1 are corrected in the final result by employing the voting mechanism.

Similar to that in Case-I, the performance of each BiLSTM classifier is evaluated and listed in Tab. 6.5. It shows high classification accuracy and verifies the effectiveness of BiLSTM classifier in sequence pattern classification.

2) Leak localization performance

In this experiment, the NPW speed estimation and leak localization results between conventional method [157] and proposed method are compared. As

Table 6.5: Classification result statistics of Case-II.

Label		*Classifier 1*	*Classifier 2*	*Classifier 3*
Leak	*Num.*	4012	4101	3746
	TPR	98.03%	97.17%	97.84%
	FDR	0.28%	0.16%	0.21%
Pump Down	*Num.*	6167	6851	6857
	TPR	99.59%	99.99%	98.56%
	FDR	0.03%	0.07%	0.1%
Pump Up	*Num.*	2885	2605	2249
	TPR	99.45%	99.23%	95.95%
	FDR	0.01%	0.01%	0.03%
Normal	*Num.*	86939	86443	87148
	TPR	99.65%	99.74%	99.74%
	FDR	0.86%	0.97%	1.4%
Accuracy		99.58%	99.64%	99.50%

shown in Tab. 6.6, the NPW speed estimated by identified pressure disturbances are listed sequentially in time order. The leak locations by proposed method and conventional method are also listed for comparison. Since the data is acquired from real industrial site and there is no prior knowledge of NPW speed, so the speed estimation comparison is not performed. However, it can be observed that after the average pipeline pressure increases at *15:18pm*, the estimated NPW speed significantly rises up accordingly. If using an assumed constant NPW speed as in the conventional method, this variation will be ignored, hence, result in poor localization accuracy.

By viewing the localization comparison, it shows the average absolute error of the proposed method is $106.8m$ which is less than that of $235m$ by the conventional method. Notably, the average localization results of the 10 experiments by the proposed method is $4959m$ which is only $41m$ different from the actual value of $5000m$. The ratios of the average error and average absolute error to the pipeline length $14.79km$ are calculated as 0.28% and 0.72% respectively, which are 0.85% and 1.59%, respectively, by using the conventional method.

Table 6.6: NPW speed and leak localization result of Case-II.

Index	*Event*	*Time*	*vel.(m/s)*	*Localization (m)*		*Absolute Error (m)*	
				Proposed	*Con.*	**Proposed**	*Con.*
1	*Pump down*	12 : 43	1045	− − −	− − −	− − −	− − −
2	*Pump up*	12 : 50	1059	− − −	− − −	− − −	− − −
3	*Leak*	12 : 59	1052	5159.5	4845	159.5	155
4	*Pump down*	13 : 08	1036	− − −	− − −	− − −	− − −
5	*Pump up*	13 : 17	1049	− − −	− − −	− − −	− − −
6	*Leak*	13 : 30	1062	5085	4785	85.15	215
7	*Pump down*	13 : 38	1031	− − −	− − −	− − −	− − −
8	*Leak*	14 : 10	1076	4893	4605	106.7	395
9	*Leak*	14 : 26	1027	5084	4695	84.3	305
10	*Leak*	14 : 49	1085	4873	4605	127	395
11	*Leak*	15 : 13	1120	4847	4665	153.5	335
12	*Pump up*	15 : 18	1195	− − −	− − −	− − −	− − −
13	*Leak*	15 : 35	1312	4869	5085	131	85
14	*Leak*	15 : 57	1324	4978	5205	22	205
15	*Leak*	16 : 20	1332	4898	5145	102	145
16	*Leak*	16 : 40	1312	4903	5115	97.2	115
Average				4959	4875	**106.8**	**235**

The experimental result of Case-II verifies the effectiveness of proposed method in real industrial scenarios. Based on the experiments, the proposed disturbance assisted method significantly outperforms the traditional constant NPW speed based method in leak localization accuracy.

6.4.3 Leak Detectability Performance

In this experiment, the leak detectability which measures the performance of the fault detector relative to the amount of fault information (i.e. leak flow) [165] is evaluated. Totally, 120 leak events are simulated under different additive white Gaussian (AWG) noise situations. The signal-to-noise ratio (SNR) is ranging from $18dB$ to $45dB$ with an increment of $3dB$ as shown in Tab. 6.7.

Tab. 6.8 shows the averaged minimum detectable ratios between leak flow

Table 6.7: Simulation environment of detectability.

	Flow(l/s)	*Number*	*Flow(l/s)*	*Number*	*Total*
Leaks	$0.11 - 4.98$	30	$4.98 - 9.43$	30	120
	$9.43 - 13.44$	30	$13.44 - 17.10$	30	
Normal Flow	$20.38l/s$	*SNR*	$18dB - 45dB$	*Step*	$3dB$
Signal variance	$895.43m^2$		*noise variance*	$0.57m^2 - 286.44m^2$	

Q_{leak} and normal flow Q in percentage form, $Q_{leak}/Q \times 100\%$. In this simulation, the normal pipe flow is $Q = 20.38l/s$. Since in practice, the employed BiLSTM classifier may not be trained under the same SNR of the contaminated pressure wave being detected, the detectabilities of a certain BiLSTM classifier under different SNRs are evaluated and listed in the columns. For example, the value "2.12" located in the column with $SNR = 21$ and the row with $SNR = 42$ means that when the leak flow is greater than 2.12% of the normal flow, it can be successfully detected by the corresponding BiLSTM classifier being trained under $SNR = 21$ and tested for detection under $SNR = 42$. The diagonal entries (marked in blue) indicate the detectability of each classifier under the same SNR condition in both training and testing. The bottom row (marked in red) shows the average detectability of each classifier under multiple noise situations. Additionally, the last column shows the average detectability with respect to multiple classifiers. It can be seen that the performance of detectability drops as the noise increases (or the SNR drops). From the last row, it can be obtained that, under the considered SNR ($18 - 45dB$), the average leak flow to be successfully detected by the BiLSTM classifier is about 4% of the normal flow, which shows a satisfactory detectability of the proposed scheme. Among all classifiers, the one trained under $30dB$ achieves the best detectability.

Table 6.8: Detectability in terms of percentage of Q_{leak}/Q under different noise conditions.

		BiLSTM classifiers										
	SNR	45	42	39	36	33	30	27	24	21	18	Ave.
Pressure waves with noises	45	**0.66**	1.20	0.95	1.32	1.64	0.99	1.63	1.21	2.47	2.55	1.46
	42	1.04	**1.42**	1.08	1.28	1.79	1.03	1.77	1.29	2.12	2.41	1.52
	39	1.10	1.74	**1.47**	1.38	1.90	1.13	2	1.52	2.39	2.34	1.70
	36	1.53	2.16	1.84	**1.70**	2.08	1.29	2.36	1.78	2.63	2.51	1.99
	33	2.41	3.17	2.77	2.25	**2.22**	1.62	2.88	2.32	3.49	3.15	2.63
	30	3.53	4.52	3.85	2.86	3.50	**2.65**	3.71	2.86	4.47	3.96	3.59
	27	5.71	5.98	5.46	4.01	4.37	2.92	**4.83**	3.82	6.73	4.94	4.88
	24	8.41	9.31	7.56	5.55	5.95	3.87	6.67	**5.12**	9.24	7.04	6.87
	21	11.72	12.81	10.16	6.58	8.19	4.64	8.63	6.49	**10.43**	9.03	8.87
	18	17.01	18.52	15.27	9.6	11.23	6.46	11.38	8.94	11.39	**11.12**	12.09
Ave.		5.31	6.08	5.04	3.65	4.29	**2.66**	4.59	3.53	5.54	4.90	

6.5 Conclusion

In this chapter, a deep learning based pipeline leak detection and localization method is proposed which can accurately detect leak events and estimate its location in multiple sensors scenarios. Compared to traditional constant NPW speed method, the proposed disturbance assisted method significantly increases the localization accuracy. Besides, the proposed method achieves high detection accuracy and the detectability is evaluated under different scales of noises.

The effectiveness of proposed method may be limited when non-leak disturbances are rare or highly randomized, and in this case, the localization performance may be comparable to the traditional constant NPW speed based method. In the future research, more applications of the proposed method in pipeline networks are to be developed. The performance is to be evaluated under more complex pipeline network topologies.

Chapter 7

Data Fusion Based Transmission Reduction and Leak Monitoring in Pipeline Networks*

7.1 Introduction

Pipeline networks transportation is the dominant measure to convey fluid industrial products such as oil, water and chemical liquids etc. They can be utilized to continuously delivering products over long distances, covering vast areas, and have become the most efficient and convenient choice in oil and water transportation. However, the long distance and vast range of pipeline distribution may incur challenges for the leak monitoring and diagnosis.

WSNs consisting a great number of low-cost sensor nodes provide efficient solutions to pipeline monitoring and leak diagnosis system [166]-[168]. Although the data redundancy introduced by multiple sensors is beneficial to leak detection and isolation, the excessive data transmissions may deplete the limited power storage of sensors in a short period of time, thus, shorten life-span of the entire WSNs based monitoring system [169]. To reduce data transmission and extend the active WSNs duration, various protocols and data fusion methods have been proposed [170]. For example, sensor nodes can be

*A version of this chapter has been submitted as "Combined dual-prediction based data fusion and enhanced leak detection and isolation method in WSN pipeline monitoring system," for publication in *IEEE Transactions on Automation Science and Engineering.*

grouped into different clusters based on their spatial distribution, with a dedicated sink node assigned to each cluster. Dual-prediction based data fusion is proposed to implement data prediction synchronously at both sensor node and the sink node in [171]-[173]. The data transmission can be skipped if the prediction error is within an acceptable range. In [174], the dual-prediction data fusion method is applied to acquire indoor environmental data in a WSNs system. The prediction model established for each sensor is independent, without considering cross-correlations with neighboring sensors. However, as shown in [175], in fluid pipeline systems, the measurement of one sensor is not only temporally auto-correlated to its historic data but also spatially cross-correlated to the measurements of its neighboring sensors. Therefore, when applying prediction based data fusion to WSNs assisted pipeline monitoring systems, prediction accuracy is expected to improve if the predictor is constructed by utilizing measurements from both of the host and neighboring sensors.

In view of this point, in this chapter, a combined dual-prediction based data fusion (CDPDF) method is proposed. In this scheme, a predictor is established based on the collections from multiple neighboring sensors instead of the traditional self-collection method. For this predictor, various machine learning based time-series forecast algorithms can be implemented, such as LSTM [176]-[179], SVM [180]-[182] and ELM [183]-[186] etc. Among those, ELM has demonstrated fast learning speed and satisfactory approximation ability, which makes it a good choice for online deployment.

In the area of leak detection and isolation, model based methods have been under continuous investigation and development due to their accuracy in describing the pipeline structure and fluid transportation dynamics. Most of the studies are based on the dynamic fluid model [187], [188] which allows to calculate the flow parameters in different parts of a pipe, including leak size and location when it happens. Various approaches have been proposed ranging from single leak to multiple leaks detection [189, 190]. Kalman filter has been widely used to perform leak detection and isolation. In [191], Extended Kalman Filter (EKF) is employed as an adaptive observer to estimate the leak parameters in an improved pipe model, where the leak size and location are taken as system states to be estimated. In [192], multi-leak situations are investigated in oil pipeline system where the Unscent Kalman Filter (UKF) is employed. It is shown that multiple leaks can be effectively detected by accordingly increase the order of observer once a new leak is detected. In [193], the Distributed Kalman Filter (DKF) method is applied to monitor leaks in the pipeline networks which extends the application of model based method to the more practical networks structure. However, in the real working condition of pipeline networks, the flow rate and pressure head in each pipe may be different and are changing over time, it would result in different time varying pipeline transient models. The aforementioned methods consider the nonlinear pipeline model as time invariant and employ the model parameters with constant empirical values, e.g., friction factor f and pressure wave propagation speed b. It will lead to inaccurate state estimation due to poor adoption of model parameters. If such model parameters can be estimated and updated online (periodically), the more accurate leak detection and isolation result can be expected. Inspired by this point, in this chapter, an enhanced leak detection and isolation (EnLDI) method is proposed which online updates the time-varying model parameters and improves the model approximation.

The main contributions are summarized as follows:

- The CDPDF method is proposed for WSNs pipeline networks monitoring system. It is compatible with a variety of supervised time-series predictors.

- The EnLDI method is proposed which can perform online updating of the time varying pipeline model parameters, while estimating the location and size of the leak in real-time.

The remainder of this chapter is organized as follows. Section 7.2 briefly reviews the fundamental principles of WSNs monitoring fluid pipeline networks and pipeline model description. Section 7.3 presents the details of the proposed method. Section 7.4 includes the experiment and comparison results with discussions. Finally, conclusion is drawn in Section 7.5.

7.2 Preliminaries

7.2.1 WSN Monitored Fluid Pipeline Networks

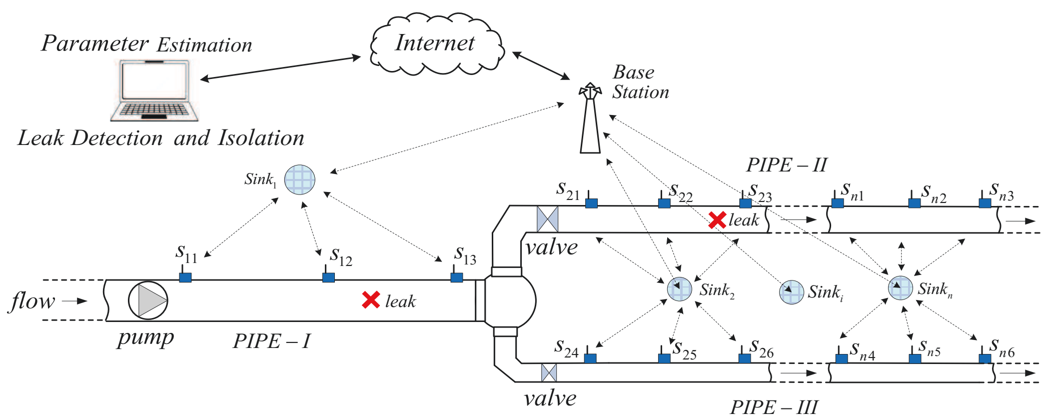

Figure 7.1: Example of the application scenario.

In a centralized WSNs data acquisition system [194], the distributed sensors are grouped into multiple clusters where each cluster has a sink node. The sink node usually owns stronger communication and calculation ability than the other children nodes within its cluster. It can communicate with all the children nodes, e.g., receive data from the distributed sensors and broadcasts instructions to those within the same cluster. Measurements collected by the distributed sensors are firstly transmitted to the sink node and then further forwarded to the base station or cloud server. Considering that the children sensor nodes are usually limited in computation and communication capabilities, in this work, we assume that the broadcast by each child sensor can at least be received by the sink node and its neighboring sensor nodes.

An example of fluid pipeline networks monitored by WSNs is shown in Fig. 7.1. There are 3 pipelines of different sizes connected and mounted with wireless sensors. These sensors measure local pressure and flow values, and are clustered based on the transmission range and pipeline topology. In this example, the sensors on PIPE-I are clustered into one group with sink node $Sink_1$, and sensors on PIPE-II and PIPE-III are clustered into multiple groups with sink nodes $Sink_2$ to $Sink_n$. Each sensor can communicate with its neighbors and the corresponding sink node, and the group measurements are transmitted to "Base Station" via the sink nodes. Since the children nodes have limited power storage, continuous data transmission between children nodes and sink node may deplete the power storage in a short term. Hence, data fusion methods that can decrease redundant transmissions play an important role in extending the WSN's active life-span. After the collected measurements are transmitted to the base station, algorithms such as network model parameter estimation, leak detection and localization etc., can be performed as part of a decision support system.

7.2.2 Pipeline Model

Given a pipe of constant cross-sectional area, assuming convective changes in velocity is negligible, the motion and continuity equations [188] governing one-dimensional dynamic fluid flow can be written as

$$\frac{\partial Q}{\partial t} + gA\frac{\partial H}{\partial z} + \mu Q|Q| = 0, \tag{7.1}$$

$$b^2\frac{\partial Q}{\partial z} + gA\frac{\partial H}{\partial t} = 0, \tag{7.2}$$

where t and z are coordinates of time (s) and space (m) respectively. H is the pressure head (m), Q is the flow (m^3/s), g is the acceleration of gravity (m/s^2), A is the cross-section area (m^2), b is the speed of pressure wave propagation (m/s) and $\mu = f/(2DA)$, where f is the friction coefficient, D is pipe diameter (m). In model-based leak detection, the leak effects can be considered as the form of flow modification. Suppose at the location p_i a leak happens, the flow at p_i can be written as:

$$Q^b_{p_i-\epsilon} = Q^a_{p_i+\epsilon} + Q_{pi} = Q^a_{p_i+\epsilon} + \lambda_i\sqrt{H_{pi}} \tag{7.3}$$

with $\epsilon \to 0$, $Q^b_{p_i-\epsilon}$ and $Q^a_{p_i+\epsilon}$ are the flows before and after the leak position p_i. Q_{pi} is the leak outflow and it can be expressed by the product of a constant λ_i and the square root of the pressure head at leak point H_{pi}, where $\lambda_i > 0$ is a function of the orifice area and discharge coefficient.

As shown in Fig. 7.2, The pipe length can be divided into n sections, $\Delta z_i, \forall i = 1, \cdots, n$. Δz_i can be different for non-uniform sectioning. To obtain a finite-dimensional description of Eqn. (7.1) and Eqn. (7.2), finite difference approach is applied as follows,

$$\frac{\partial H_i}{\partial z} \simeq \frac{H_{i+1} - H_i}{\Delta z_i} \quad \forall i = 1, \cdots, n \tag{7.4}$$

$$\frac{\partial Q_i}{\partial z} \simeq \frac{Q_i - Q_{i-1}}{\Delta z_i} \quad \forall i = 2, \cdots, n \tag{7.5}$$

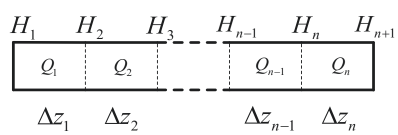

Figure 7.2: Pipe discretization in space.

Then, the nominal pipeline model can be obtained as n sets of coupled nonlinear differential equations given by

$$\frac{\partial Q_i}{\partial t} = -\mu Q_i|Q_i| - \frac{gA}{\Delta z_i}(H_{i+1} - H_i) \quad \forall i = 1, \cdots, n \tag{7.6}$$

$$\frac{\partial H_i}{\partial t} = \frac{b^2}{\Delta z_i gA}(Q_{i-1} - \lambda_{i-1}\sqrt{H_i} - Q_i) \quad \forall i = 2, \cdots, n \tag{7.7}$$

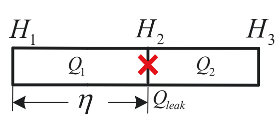

Figure 7.3: A section of pipe with one leak.

For simplicity but without loss of generality, we consider the case of single leak in a section of pipeline of length L, as shown in Fig. 7.3. Assume that at both ends, the pressure heads and flow rates are measured, denoted as H_1, Q_1 and H_3, Q_2, respectively. Assuming that the leak occurs at the location η

and the leak flow is $Q_{leak} = \lambda_1 \sqrt{H_2}$, the pipeline model that incorporates the leak parameters (leak location and leak size) can then be established as

$$\begin{bmatrix} \dot{Q}_1 \\ \dot{H}_2 \\ \dot{Q}_2 \\ \dot{\eta} \\ \dot{\lambda}_1 \end{bmatrix} = \begin{bmatrix} -\mu Q_1 |Q_1| + \frac{gA}{\eta}(H_1 - H_2) \\ \frac{b^2}{gA(L-\eta)}(Q_1 - \lambda_1 \sqrt{H_2} - Q_2) \\ -\mu Q_2 |Q_2| + \frac{gA}{L-\eta}(H_2 - H_3) \\ 0 \\ 0 \end{bmatrix} \tag{7.8}$$

7.3 Proposed Method

As explained in Sec. 7.2.1, to reduce the number of unnecessary wireless transmissions and prolong the lifespan of WSNs, a CDPDF model is first introduced. The measured pressure and flow are fused with the predicted values in this model. Then using fused data, the EnLDI is designed based on an adaptive extended Kalman Filter.

7.3.1 Combined Dual-prediction Based Data Fusion

The data transmission part of the WSNs based pipeline monitoring system shown in Fig. 7.1 can be further depicted in Fig. 7.4. Take the i-th cluster as an example, where m children nodes S_1 to S_m are consisted in this cluster and the sink node is $Sink_i$. Before detailed introduction, assumptions and terms of proposed CDPDF are reiterated and clarified as follows.

- The broadcast by $Sink_i$ can be received by all its children nodes, and the broadcast by a child node S_j can at least be received by the sink node $Sink_i$ and the neighbored children nodes S_{j-1} and S_{j+1}.
- Considering the limited power and calculation capability of children nodes, the training process of the proposed predictor is implemented only at the sink node, e.g., $Sink_i$.
- The term "combined" of the proposed method means that the prediction is made by combining the measurement collection of the local node and its neighboring sensor nodes.

- The term "dual-prediction" means that the prediction is synchronously conducted at both $Sink_i$ and its corresponding child node S_j.

- Data fusion in this work is mainly for reducing wireless transmissions between the sink nodes and children nodes. Data transmission is only triggered if the predicted value is substantially different from the measured one. The final data sequence obtained by sink node is fused with both predicted and measured values. The more predicted values it adopts, the less wireless transmissions are demanded.

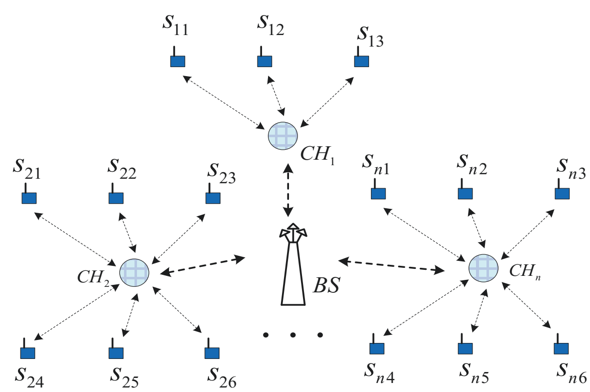

Figure 7.4: Example of sensor formation.

1) Procedures of proposed CDPDF method

To save redundant transmissions between the sink node and its children nodes, the proposed CDPDF include four main phases in each cycle, namely, training, synchronized prediction, fusion, and updating. By taking the i-th sink and j-th child sensor node as an example, design details are elaborated as follows.

Training:

1. On the sink side, $Sink_i$ broadcasts initial instruction to the children nodes in cluster i.

2. On the children nodes side, each of the children nodes in cluster i begins to collect $N+1$ initial data points upon receiving the instruction. Given the j-th child node S_j, the initial fused sequence is set as the initial collection, denoted as

$$\begin{aligned} \mathbf{X}_j^f &= [x_j(1), \cdots, x_j(N+1)] \\ &= [x_j^f(1), \cdots, x_j^f(N+1)], \quad j = 1, \cdots, m, \end{aligned}$$

where the superscript f denotes the fused sequence and m is the population in cluster i.

3. Children nodes transmit all the initial fused sequences $\mathbf{X}_j^f, j = 1, \cdots, m$ to $Sink_i$, and the broadcast can also be received by the neighboring children nodes. For example, S_j can receive the initial fused sequences $\mathbf{X}_{j-1}^f$ and $\mathbf{X}_{j+1}^f$ from its neighbor nodes S_{j-1} and S_{j+1}.

4. On the sink side, $Sink_i$ collects the initial training data sets $\left[\boldsymbol{X}_j, \boldsymbol{Y}_j\right]$ for the predictor $\boldsymbol{F}_j(\cdot)$, where $\boldsymbol{F}_j(\cdot)$ is the one-step ahead predictor of the measurement for child node S_j. The initial training set for $\boldsymbol{F}_j(\cdot)$ is set as

$$\begin{aligned} \boldsymbol{X}_j &= \begin{bmatrix} \mathbf{X}_{j-1}^f \\ \mathbf{X}_j^f \\ \mathbf{X}_{j+1}^f \end{bmatrix} = \begin{bmatrix} x_{j-1}^f(1), \cdots, x_{j-1}^f(N) \\ x_j^f(1), \cdots, x_j^f(N) \\ x_{j+1}^f(1), \cdots, x_{j+1}^f(N) \end{bmatrix} \\ \boldsymbol{Y}_j &= \left[x_j^f(2), \cdots, x_j^f(N+1)\right] \end{aligned}$$

It can be seen that the collection from S_j and its neighboring sensors S_{j-1} and S_{j+1} are combined into the training set $\boldsymbol{X}_j$. Therefore, not only auto-correlation of the data from a single node S_j but the spatial cross-correlation among neighboring sensors are also exploited to build the predictor.

5. $Sink_i$ applies supervised learning method to train m extreme learning machine (ELM) based predictors $\left[\boldsymbol{F}_1(\cdot), \cdots, \boldsymbol{F}_m(\cdot)\right]$ with their corresponding training set $\left[\boldsymbol{X}_j, \boldsymbol{Y}_j\right], j = 1, \cdots, m$. The detailed training process based on ELM is introduced in the next section.

Synchronized Prediction:

6. On the sink side, $Sink_i$ transmits the parameters of trained predictors to the children nodes. Each child node receives predictor parameters not only for itself, but also for its neighbors. For example, child node S_j receives the predictors as $\left[\boldsymbol{F}_{j-1}(\cdot), \boldsymbol{F}_j(\cdot), \boldsymbol{F}_{j+1}(\cdot)\right]$. Thus, m predictors are constructed at the sink node and 3 predictors are obtained by each of the children nodes.

7. On both the sink and children nodes, $Sink_i$ and S_j, the k-th step prediction are performed synchronously, $k > N+1$,

$$\begin{aligned} \hat{x}_{j-1}(k) &= \boldsymbol{F}_{j-1}\left(x^f_{j-1}(k-1), x^f_j(k-1), x^f_{j+1}(k-1)\right) \\ \hat{x}_j(k) &= \boldsymbol{F}_j\left(x^f_{j-1}(k-1), x^f_j(k-1), x^f_{j+1}(k-1)\right) \\ \hat{x}_{j+1}(k) &= \boldsymbol{F}_{j+1}\left(x^f_{j-1}(k-1), x^f_j(k-1), x^f_{j+1}(k-1)\right) \end{aligned}$$

Transmission and Fusion:

8. On the children nodes side, S_j acquires new measurement $x_j(k)$, and an error threshold $\boldsymbol{\epsilon}$ is selected. When the prediction error is within the threshold $\boldsymbol{\epsilon}$, no transmission of the new measurement from S_j to $Sink_i$ is needed and both sides will accept the predicted value. Otherwise, if the prediction error is beyond $\boldsymbol{\epsilon}$, S_j transmits $x_j(k)$ to $Sink_i$ and the transmitted measurement is put in the fused data sequence $\mathbf{X}^f_j$,

$$\begin{aligned} \mathbf{X}^f_j &= [x^f_j(1), \cdots, x^f_j(N+1), \cdots, \hat{x}_j(k)] \qquad (7.9) \\ s.t. \quad & |x_j(k) - \hat{x}_j(k)| \leq \boldsymbol{\epsilon}, \quad j = 1, \cdots, m. \\ \mathbf{X}^f_j &= [x^f_j(1), \cdots, x^f_j(N+1), \cdots, x_j(k)] \qquad (7.10) \\ s.t. \quad & |x_j(k) - \hat{x}_j(k)| > \boldsymbol{\epsilon} \quad j = 1, \cdots, m. \end{aligned}$$

9. On the children nodes side, if S_j receives real measurement $x_{j-1}(k)$ or $x_{j+1}(k)$ transmitted from S_{j-1} or S_{j+1}, the received value will be taken as next-step input of $\boldsymbol{F}_j(\cdot)$, otherwise, the predicted $\hat{x}_{j-1}(k)$ and $\hat{x}_{j+1}(k)$ will be taken as the next-step input of predictor $\boldsymbol{F}_j(\cdot)$.

Updating:

10. On the sink side, if a transmission occurs in step 8, $Sink_i$ starts a new round of predictor training with the training data set $\left[\boldsymbol{X}_j, \boldsymbol{Y}_j\right]$ as follows,

$$\boldsymbol{X}_j = \begin{bmatrix} x^f_{j-1}(k-N), \cdots, x^f_{j-1}(k-1) \\ x^f_{j}(k-N), \cdots, x^f_{j}(k-1) \\ x^f_{j+1}(k-N), \cdots, x^f_{j+1}(k-1) \end{bmatrix} \tag{7.11}$$

$$\boldsymbol{Y}_j = [x^f_j(k-N+1), \cdots, x^f_j(k)] \tag{7.12}$$

Once the new predictor $\boldsymbol{F}_j(\cdot)$ is trained, its updated parameters are transmitted to S_{j-1}, S_j and S_{j+1} to replace the previous ones.

11. Proceed to step 7 and continue next step prediction on both sink and children nodes sides.

2) The construction of ELM based predictor

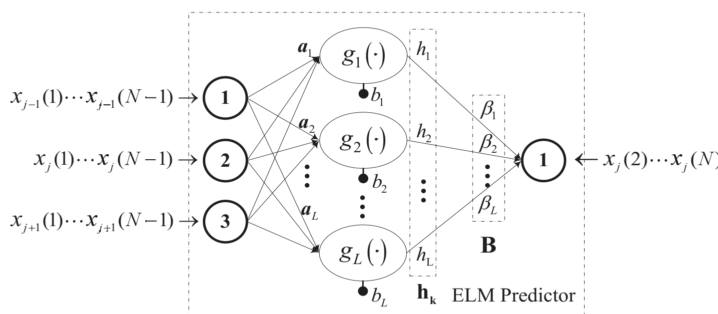

Figure 7.5: Structure of ELM predictor.

It is known that the extreme learning machine operates as a generalized single hidden layer feed-forward neural network with no need to tune parameters of the hidden layer. As proposed in [97], those parameters (i.e., input weights and biases of the hidden nodes) can be randomly assigned by following any continuous distribution function and then remain fixed for calculating the output layer weights afterwards. The output weights are analytically determined by least-squares solutions of a general system of linear equations [97]. Generally speaking, the ELM operates as an universal approximator, which renders superior approximation performance in most cases and can learn thousands of times faster than other conventional algorithms due to its computational simplicity. Those advantages make it a desirable choice for online time-series forecaster.

The proposed structure for one of the m predictors in the i-th cluster is shown in Fig. 7.5 and the subscript j ($j = 1, \cdots, m$) is the index of sensor node within cluster i, $i = 1, \cdots, n$. The three layers of the proposed ELM predictor are described as follows.

- Input layer: The 3 nodes of the input layer are corresponding to the fused data vector of sensor S_j and its neighbor sensors S_{j-1} and S_{j+1} at time instant k, denoted as

$$\mathbf{X}_j^f(k) = [x_{j-1}^f(k), x_j^f(k), x_{j+1}^f(k)]^T, \tag{7.13}$$

- Hidden layer: The hidden layer contains L hidden nodes with activation function $g(x) : R^3 \to R$. The output of the l-th hidden node is given by

$$h_l = g(\boldsymbol{a}_l{}^T \cdot \mathbf{X}_j^f(k) + b_l), \tag{7.14}$$

  where $l = 1, \cdots, L$ and $\boldsymbol{a}_l \in R^3$ is the column vector of weights connecting the l-th hidden node and the three input nodes. b_l is the bias of the l-th hidden node. Hereby $\boldsymbol{a}_l$ and b_l are randomly assigned following uniform distribution on $[-0.2 \quad 0.2]$ and keep fixed afterwards.

- Output layer: To operate as a single-value predictor, the output layer contains only 1 node corresponding to the one step ahead predicted value $\hat{x}_j(k)$. Then, the output of an ELM network with L hidden nodes can be written by

$$\hat{x}_j(k) = \sum_{l=1}^{L} h_l \cdot \beta_l = \boldsymbol{h}_k \boldsymbol{B}, \quad \beta_l \in R, \tag{7.15}$$

  where $\boldsymbol{h}_k$ is the row vector of the L hidden nodes outputs when input is the $\boldsymbol{X}_j^f(k)$. $\boldsymbol{B}$ is the $L \times 1$ output weights vector connecting hidden layer and output layer. $\boldsymbol{B}$ can be written as $\boldsymbol{B} = [\beta_1 \cdots \beta_l \cdots \beta_L]^T$. The output weights matrix $\boldsymbol{B}$ can be obtained by training the network, which will be explained in the following.

As explained in the above procedure, all training is performed at the sink node. To perform training of the ELM predictor of sensor j at time instant $k \geq N+1$, the training data set is chosen as $[\boldsymbol{X}_j, \boldsymbol{Y}_j]$, where $\boldsymbol{X}_j$ and $\boldsymbol{Y}_j$ are shown in Eqn. (7.11) and Eqn. (7.12). $\boldsymbol{X}_j$ is the fused data matrix of S_j and its neighbors for N continuous instants from time instant $k-N$ to $k-1$. Rewrite $\boldsymbol{X}_j$ in its column form as follows.

$$\boldsymbol{X}_j = [X_1, \cdots, X_p, \cdots, X_N]. \tag{7.16}$$

and the training target $\boldsymbol{Y}_j$ is selected as one step ahead time sequence of its corresponding measurements.

Let

$$\boldsymbol{H}_j = \begin{bmatrix} \boldsymbol{h}_1 \\ \vdots \\ \boldsymbol{h}_p \\ \vdots \\ \boldsymbol{h}_N \end{bmatrix} = \begin{bmatrix} g(\boldsymbol{a}_1^T X_1 + b_1) \cdots g(\boldsymbol{a}_L^T X_1 + b_L) \\ \vdots \qquad \cdots \qquad \vdots \\ g(\boldsymbol{a}_1^T X_p + b_1) \cdots g(\boldsymbol{a}_L^T X_p + b_L) \\ \vdots \qquad \cdots \qquad \vdots \\ g(\boldsymbol{a}_1^T X_N + b_1) \cdots g(\boldsymbol{a}_L^T X_N + b_L) \end{bmatrix} \tag{7.17}$$

$\boldsymbol{H}_j$ denotes the $N \times L$ hidden layer outputs matrix in the training phase of predictor j. The N rows of $\boldsymbol{H}_j$ are the hidden layer output vectors corresponding to the N inputs in $\boldsymbol{X}_j$ during training.

Establish the cost function which considers the empirical and structural risk of the proposed ELM as:

$$Minimize : \mathcal{L}_{ELM} = \frac{1}{2}||\boldsymbol{B}||^2 + \frac{C}{2}||\boldsymbol{Y}_j - \boldsymbol{H}_j\boldsymbol{B}||^2 \tag{7.18}$$

Solving the regularized least squares optimization problem in (7.18), we have

$$\boldsymbol{B} = \left(\boldsymbol{H}_j^T \boldsymbol{H}_j + \frac{\boldsymbol{I}_L}{C} \right)^{-1} \boldsymbol{H}_j^T \boldsymbol{Y}_j, \tag{7.19}$$

where C is the coefficient for balancing the empirical and structural risks, $\boldsymbol{I}_L$ is the L dimensional identity matrix. Once the training is performed and $\boldsymbol{B}$ obtained, the ELM based predictor is constructed. The sink node will then transmit the parameters such as input weights, hidden neuron biases and output weights $\boldsymbol{B}$ to all children nodes within the cluster to perform the synchronized prediction.

Remark 1. Instead of using ELM, other approaches in machine learning can also be applied to construct the predictors, such as SVM, LSTM, etc. The main reason we adopt and present the ELM for the predictors is because ELM has much less computational complexity compared with other approaches.

Remark 2. Considering the limited calculation capacity of children nodes, only prediction related calculations such as Eqn. (7.14) and Eqn. (7.15) are implemented in children nodes. The predictor training calculation of Eqn. (7.19) is implemented in sink node which owns stronger calculation capability.

7.3.2 Enhanced Pipeline Leak Detection and Isolation Method

The collected pressure and flow data sequences are sent to the base station via WSNs. In this work, we propose an online leak detection and isolation scheme based on adaptive extended Kalman filter.

1) Online pipeline model parameters estimation

When no leak occurs the fluid dynamic equations of Eqn. (7.6) and Eqn. (7.7) can be written in the forward difference equation form (with sampling time Δt) as:

$$\begin{aligned} Q_i(k+1) - Q_i(k) - \frac{gA}{\Delta z_i}(H_i(k) - H_{i+1}(k))\Delta t = \\ -\mu Q_i(k)|Q_i(k)|\Delta t \end{aligned} \tag{7.20}$$

$$H_i(k+1) - H_i(k) = \frac{b^2}{gA\Delta z_i}(Q_{i-1}(k) - Q_i(k))\Delta t \tag{7.21}$$

In practice, the values of friction f (hence $\mu = \frac{f}{2DA}$) and pressure wave propagation speed b can be varying under different conditions such as fluid viscosity and density, pipe roughness and pressure etc. Consider the first two sections of pipeline shown in Fig. 7.2, Eqn. (7.20) and Eqn. (7.21) are written as:

$$\begin{aligned} Q_1(k+1) - Q_1(k) - \frac{gA}{\Delta z_1}(H_1(k) - H_2(k))\Delta t = \\ -\mu Q_1(k)|Q_1(k)|\Delta t \end{aligned} \tag{7.22}$$

$$H_2(k+1) - H_2(k) = \frac{b^2}{gA\Delta z_2}(Q_1(k) - Q_2(k))\Delta t \tag{7.23}$$

With the measurements at both ends H_1, H_2 and Q_1, Q_2, the time-varying model parameters μ and b can be estimated.

In Eqn. (7.22), define

$$\Phi(k+1) = \begin{bmatrix} Q_1(k+1) - Q_1(k) - \frac{gA}{\Delta z_1}(H_1(k) - H_2(k))\Delta t \\ H_2(k+1) - H_2(k) \end{bmatrix} \tag{7.24}$$

and

$$M(k) = \begin{bmatrix} -Q_1(k)|Q_1(k)|\Delta t & 0 \\ 0 & \frac{1}{gA\Delta z_2}(Q_1(k) - Q_2(k))\Delta t \end{bmatrix} \tag{7.25}$$

Let $\theta = \begin{bmatrix} \mu \\ b^2 \end{bmatrix}$ be the estimated, then we obtain,

$$\Phi(k+1) = M(k)\theta(k) \tag{7.26}$$

One option is to calculate $\hat{\theta} = \begin{bmatrix} \hat{\mu} \\ \hat{b}^2 \end{bmatrix}$ periodically via an ordinary least squares solution upon reception of a block of new data points at the base station. Another option is to estimate the parameters μ and b through a recursive least square (RLS) estimation scheme. By estimating these parameters, the leak detection and localization precision can be dramatically improved compared with the case when empirical constant values for μ and b are employed.

2) Leak detection and isolation based on EKF

Take the example shown in Fig. 7.3, where one leak occurs between two sensing spots where the pressure $[H_1, H_3]$ and flow $[Q_1, Q_2]$ are measured.

Define the input vector as $U = [u_1, u_2]^T$ where $u_1 = H_1$ and $u_2 = H_3$, the output vector as $Y = [y_1, y_2]^T$ where $y_1 = Q_1$ and $y_2 = Q_2$. The system state vector is defined as:

$$\boldsymbol{x} = [x_1, x_2, x_3, x_4, x_5]^T = [Q_1, H_2, Q_2, \eta, \lambda_1]^T \tag{7.27}$$

Particularly $x_2 = H_2$, the leak spot pressure, $x_4 = \eta$, the distance from the left sensor to the leak spot, and $x_5 = \lambda_1$, the leak size, are of most interests.

The state space model of Eqn. (7.8) with estimated parameters $\hat{\mu}$ and $\hat{b}$ can be rewritten as:

$$\begin{aligned}
\dot{\boldsymbol{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \end{bmatrix} &= \begin{bmatrix} -\hat{\mu}x_1|x_1| + \frac{gA}{x_4}(u_1 - x_2) \\ \frac{\hat{b}^2}{gA(L-x_4)}(x_1 - x_5\sqrt{x_2} - x_3) \\ -\hat{\mu}x_3|x_3| + \frac{gA}{L-x_4}(x_2 - u_3) \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} \hat{f}_1(t, x_1, x_2, x_4, u_1) \\ \hat{f}_2(t, x_1, x_2, x_3, x_4, x_5) \\ \hat{f}_3(t, x_2, x_3, x_4, u_2) \\ 0 \\ 0 \end{bmatrix} \qquad (7.28) \\
\boldsymbol{y} &= \begin{bmatrix} 1\,0\,0\,0\,0 \\ 0\,0\,1\,0\,0 \end{bmatrix} \boldsymbol{x} = \boldsymbol{C}\boldsymbol{x} \qquad (7.29)
\end{aligned}$$

where $\hat{f}_i(\cdot)$, $i = 1, 2, 3$ represent the functions with estimated parameters.

In this work, the continuous nonlinear system model is discretized by Runge–Kutta method as following. The discretized equation of Eqn. (7.28) can be written as:

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \\ x_3(n+1) \\ x_4(n+1) \\ x_5(n+1) \end{bmatrix} = \begin{bmatrix} x_1(n) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ x_2(n) + \frac{1}{6}(l_1 + 2l_2 + 2l_3 + l_4) \\ x_3(n) + \frac{1}{6}(m_1 + 2m_2 + 2m_3 + m_4) \\ x_4(n) \\ x_5(n) \end{bmatrix}$$

for $n = 0, 1, 2, 3, \cdots$ and $t_{n+1} = t_n + h$, h is the sampling interval. The increments k_i, l_i and m_i for each of the discretized equation are as follows:

$$
\begin{cases}
k_1 = \hat{f}_1(t_n, x_1(n), x_2(n), x_4(n), u_1(n)) \\
k_2 = \hat{f}_1(\alpha_1, x_4(n), u_1(n)) \\
k_3 = \hat{f}_1(\beta_1, x_4(n), u_1(n)) \\
k_4 = \hat{f}_1(\gamma_1, x_4(n), u_1(n)) \\
\alpha_1 = \{t_n + \frac{h}{2}, x_1(n) + \frac{h}{2}k_1, x_2(n) + \frac{h}{2}l_1\} \\
\beta_1 = \{t_n + \frac{h}{2}, x_1(n) + \frac{h}{2}k_2, x_2(n) + \frac{h}{2}l_2\} \\
\gamma_1 = \{t_n + h, x_1(n) + hk_3, x_2(n) + hl_3\}
\end{cases}
$$

$$
\begin{cases}
l_1 = \hat{f}_2(t_n, x_1(n), x_2(n), x_3(n), x_4(n), x_5(n)) \\
l_2 = \hat{f}_2(\alpha_2, x_4(n), x_5(n)) \\
l_3 = \hat{f}_2(\beta_2, x_4(n), x_5(n)) \\
l_4 = \hat{f}_2(\gamma_2, x_4(n), x_5(n)) \\
\alpha_2 = \{t_n + \frac{h}{2}, x_1(n) + \frac{h}{2}k_1, x_2(n) + \frac{h}{2}l_1, x_3(n) + \frac{h}{2}m_1\} \\
\beta_2 = \{t_n + \frac{h}{2}, x_1(n) + \frac{h}{2}k_2, x_2(n) + \frac{h}{2}l_2, x_3(n) + \frac{h}{2}m_2\} \\
\gamma_2 = \{t_n + h, x_1(n) + hk_3, x_2(n) + hl_3, x_3(n) + hm_3\}
\end{cases}
$$

$$
\begin{cases}
m_1 = \hat{f}_3(t_n, x_2(n), x_3(n), x_4(n), u_2(n)) \\
m_2 = \hat{f}_3(\alpha_3, x_4(n), u_2(n)) \\
m_3 = \hat{f}_3(\beta_3, x_4(n), u_2(n)) \\
m_4 = \hat{f}_3(\gamma_3, x_4(n), u_2(n)) \\
\alpha_3 = \{t_n + \frac{h}{2}, x_2(n) + \frac{h}{2}l_1, x_3(n) + \frac{h}{2}m_1\} \\
\beta_3 = \{t_n + \frac{h}{2}, x_2(n) + \frac{h}{2}l_2, x_3(n) + \frac{h}{2}m_2\} \\
\gamma_3 = \{t_n + h, x_2(n) + hl_3, x_3(n) + hm_3\}
\end{cases}
$$

After discretization, the discrete-time model can be written in the form of backward difference in Eqn. (7.30) and Eqn. (7.31).

$$\boldsymbol{x}_k = \hat{\boldsymbol{f}}_d(\boldsymbol{x}_{k-1}, \boldsymbol{u}_{k-1}) + \boldsymbol{\omega}_k \tag{7.30}$$

$$\boldsymbol{y}_k = \boldsymbol{C}_k \boldsymbol{x}_k + \boldsymbol{v}_k \tag{7.31}$$

where $\boldsymbol{C}_k = \boldsymbol{C}$, $\hat{\boldsymbol{f}}_d(\cdot)$ are given in the Appendix. $\boldsymbol{\omega}_k$ and $\boldsymbol{v}_k$ are the process and observation errors which are both assumed to be zero mean multivariate Gaussian noises with covariance $\boldsymbol{\Sigma}_k$ and $\boldsymbol{R}_k$ respectively. $\boldsymbol{u}_k$ is the control vector. The function $\hat{\boldsymbol{f}}_d(\cdot)$ is used to predict the state from its previous measurement, denoted as $\hat{\boldsymbol{x}}_{k|k-1}$.

The main steps of EKF are given as follows:

$$\hat{\boldsymbol{x}}_{k|k} = \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k\left(\boldsymbol{y}_k - \boldsymbol{C}_k\hat{\boldsymbol{x}}_{k|k-1}\right) \tag{7.32}$$

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^T\left(\boldsymbol{C}_k\boldsymbol{P}_{k|k-1}\boldsymbol{C}_k^T + \boldsymbol{R}_k\right)^{-1} \tag{7.33}$$

$$\boldsymbol{P}_{k|k-1} = \hat{\boldsymbol{F}}_{k-1}\boldsymbol{P}_{k-1|k-1}\hat{\boldsymbol{F}}_{k-1}^T + \boldsymbol{\Sigma}_{k-1} \tag{7.34}$$

$$\boldsymbol{P}_{k|k} = (\boldsymbol{I} - \boldsymbol{K}_k\boldsymbol{C}_k)\boldsymbol{P}_{k|k-1} \tag{7.35}$$

where the state transition matrix $\hat{\boldsymbol{F}}_{k-1}$ is computed as the Jacobian with the estimated parameters $\hat{\mu}$ and $\hat{b}$ substituted in:

$$\hat{\boldsymbol{F}}_{k-1} = \left.\frac{\partial \hat{\boldsymbol{f}}_d}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}_{k-1|k-1},\boldsymbol{u}_{k-1}} \tag{7.36}$$

and $\hat{\boldsymbol{x}}_{k|k}$ is the corrected state estimation at time instant k.

Finally, by incorporating the online parameter adaptation, an improved EKF based leak detection and localization scheme can be established. When $\hat{x}_5 > \gamma$, where γ is a preset threshold, a leak is flagged and $\hat{x}_4$ will provide the estimated location for the leak.

7.4 Experiment

In this section, a pipeline network model based on a real fluid transportation scenario is used for testing, as shown in Fig. 7.6. It consists of four different sizes of pipes. Along each pipe, 3 sensors are deployed to measure local pressure head and flow rate. The corresponding pipe parameters are displayed in Tab. 7.1. The pipe diameter and wall thickness are listed in the third and fourth columns and the distances between sensors are also listed in the last two columns. This pipeline network model is constructed by using software “Allievi”.

Three main case studies are conducted. In the first one, we test the data fusion performance of proposed CDPDF with comparison results; then we evaluate the leak detection and isolation performance of proposed EnLDI method in the second case study; finally in the third case study we integrate CDPDF and EnLDI in the simulated network system to validate the overall WSNs based leak detection and localization performance.

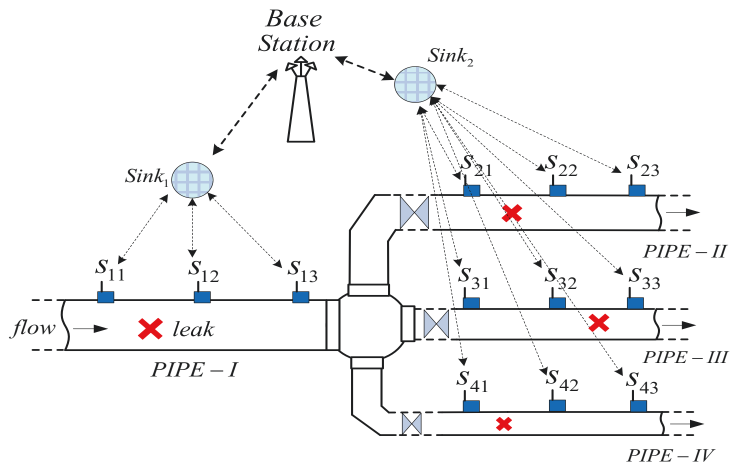

Figure 7.6: Simulation scenario.

Table 7.1: Pipe parameters of simulation.

Item	Type	Pipe(mm)		Distance(m)	
		Diam.	*Thick.*	$S_1 - S_2$	$S_2 - S_3$
I	$DN200$	219	8.18	3000	3000
II	$DN150$	168.3	7.11	2000	2000
III	$DN125$	141.3	6.55	2000	2000
IV	$DN100$	114.3	6.02	2000	2000

7.4.1 Experiment-I: Performance of Proposed CDPDF

1) Data fusion results from PIPE-I

In this experiment, 30 minutes of data is acquired by each sensor with sampling period as 1 second. Therefore, the pressure and flow with respect to 1800 continuous time instants are acquired by each sensor. In Fig. 7.7 and Fig. 7.8, the fused pressure and flow from the PIPE-I from t = 400s to t = 600s are plotted, when error tolerance thresholds are set as $\epsilon = 0.1m$ and $\epsilon = 0.1m^3/s$ respectively. By employing the proposed CDPDF with ELM based predictor, the least transmission occurs in the second subplot of Fig. 7.8

where only 192 transmissions are needed to obtained 1800 data points. The third subplot in Fig. 7.7 shows the case with most transmissions, which is 744 but still much less than the total number 1800. It can be seen that a great amount of redundant transmission are effectively eliminated.

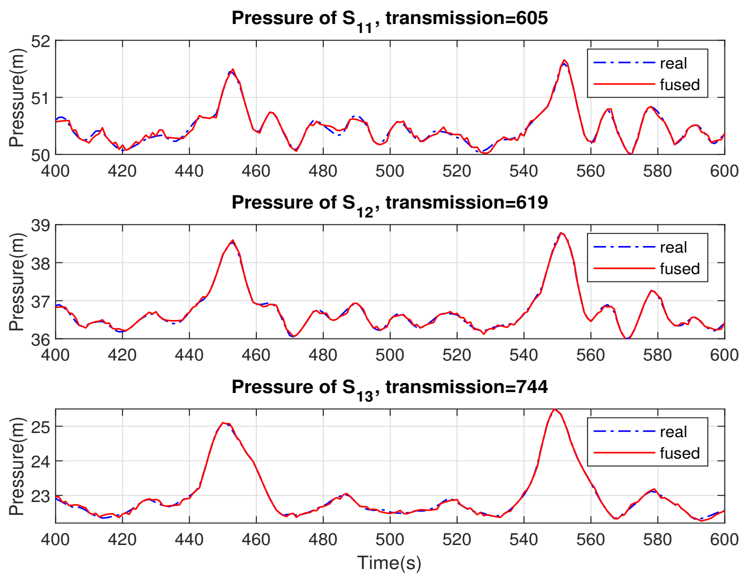

Figure 7.7: Example of fused pressure data when $\epsilon = 0.1m$.

2) Comparison results

Herein we compare the proposed *combined* prediction based data fusion with the approach which only considers the single sensor's measurements in the prediction. Furthermore, performance comparison among different machine learning based predictors is given.

Define the "*Prediction Rate*" as the ratio between the number of predictions that the errors are within error tolerance range ϵ and the number of total predictions. From Fig. 7.9, it can be seen that the "*Prediction Rate*" of all predictors are increasing when relaxing the "*Error threshold*", it suggests that while satisfying the requirement of data accuracy, choosing larger error tolerance can significantly decreases the amount of transmissions.

Also, compared with traditional single sensor methods such as *LSTM_Single*,

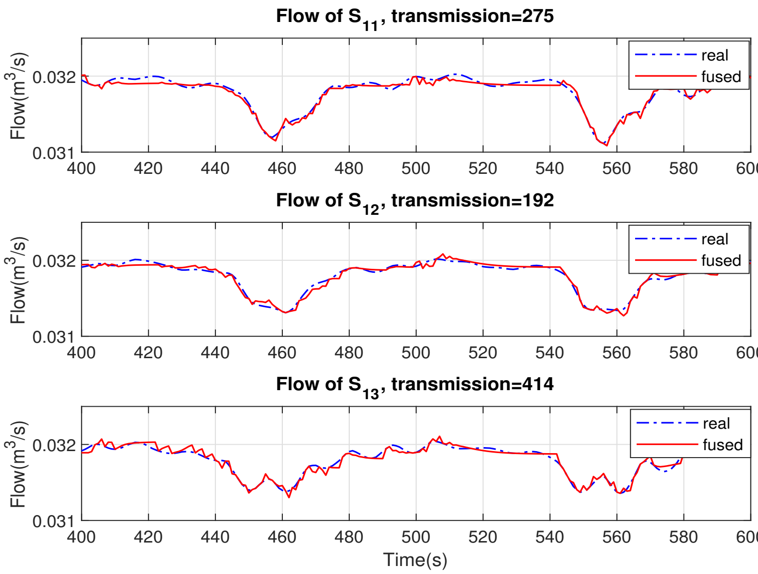

Figure 7.8: Example of fused flow data when $\epsilon = 0.1m^3/s$.

ELM_Single and *SVM_Single*, the corresponding methods under the proposed CDPDF structure e.g., *LSTM_Combined*, *ELM_Combined* and *SVM_Combined* are all achieving performance improvement in terms of higher prediction rate. It verifies that the proposed method by combining measurements from neighboring sensors outperforms the single sensor methods even in various predictor selections.

Furthermore, it can be observed that among the three implemented machine learning approaches for predictors, *LSTM* achieves the highest prediction accuracy. This benefits from its deep learning structure and memorizing ability of historical data features. *ELM* renders the second highest prediction rate. Among all three approaches, SVM has the lowest rate. But even when taking *SVM* as predictor in the proposed structure with error threshold $\epsilon = 0.1m$, it can still achieve almost 60% prediction rate, leading to significant reduction of transmissions.

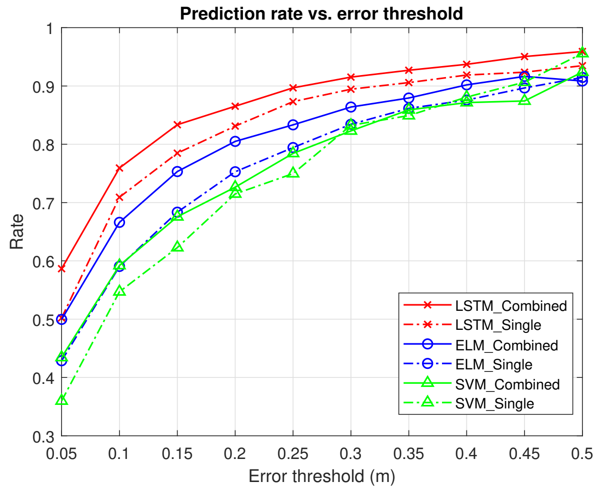

Figure 7.9: Prediction rate against Error threshold ϵ.

3) Transmission numbers and time consumed on learning

In this experiment, the transmission number of proposed CDPDF and traditional single sensor methods are compared under the setting of different ϵ. Also, the total and average time consumed on predictor model learning are compared.

Fig. 7.10 shows the transmission number when employing different predictor algorithms in the proposed method under various of ϵ settings. For the case of $\epsilon = 0.1$m, the numbers are also listed in Table 7.2. It can be seen that the method "*LSTM_Combined*" achieves least transmission numbers. When $\epsilon = 0.1m$, it only requires 410 times of transmission to acquire 1800 data points, while "*ELM_Combined*" and "*SVM_Combined*" need 568 and 694 times respectively. Although "*LSTM_Combined*" achieves the highest prediction accuracy and lowest data transmission number, it may not be suitable for time-sensitive online working situations especially when computation capacity is not sufficient for many remote wireless sensors.

The total and average time consumed on model learning are summarized In Table 7.2. The time consumption is evaluated on a computer with $i5 - 8400$ CPU and $8GB$ RAM in Windows 10 operation system. It can be seen in the blue colored text that the time consumed on $LSTM$ predictor learning is

1668.6s in total and 4.20s on average which is much greater than that of other methods. The slow learning speed makes $LSTM$ not an appropriate choice in this working condition with sampling period of 1s. On the other hand, ELM achieves much faster learning speed than other methods with 0.0014s as marked in red. Its fast learning speed, satisfactory prediction accuracy and transmission rate make it a suitable candidate in the online monitoring situation. Meanwhile, it is shown that the learning time with "*Com.*" (combined measurements from a group of sensors) is slower than that with "*Sgl.*" (single sensor measurement), because it is indeed more computationally involved to learn a model from multiple sensors than from just a single one. But the increase of time cost is moderate and can usually be handled in practice.

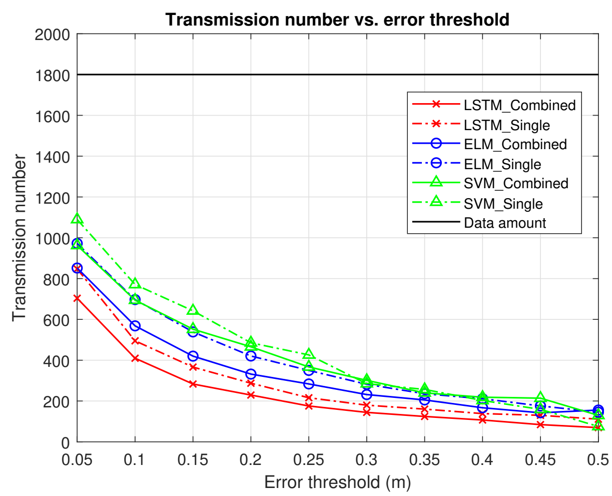

Figure 7.10: Transmission number against error threshold ϵ.

Table 7.2: Transmission and time consumption of Experiment-I.

Items	LSTM		ELM		SVM	
	Com.	*Sgl.*	*Com.*	*Sgl.*	*Com.*	*Sgl.*
Trans. number	410	495	568	694	694	771
Total time(s)	1668.6	1272.8	0.9867	0.89	5.2933	5.9967
Average time(s)	4.20	2.54	0.0014	0.0011	0.0078	0.0074

Table 7.3: Comparison of leak localization results of Experiment-II.

Index	Friction		Speed	Location(m)			Error(m)	
	True	Esti.	(m/s)	True	Con.	Pro.	Con./%	Pro./%
1	0.01858	0.01863	1206.98	2782.1	2650.9	2649.6	131.1/4.4	132.4/4.4
2	0.01745	0.01751	1224.21	2752.5	2291.3	2730.4	461.2/15.4	22.0/0.7
3	0.01736	0.01742	1225.18	2140.7	1702.8	2066.0	437.9/14.6	74.7/2.5
4	0.01749	0.01755	1223.64	1855.0	1527.9	1827.8	327.1/10.9	27.2/0.9
5	0.02665	0.02668	1141.60	1029.9	1267.1	903.6	237.3/7.9	126.3/4.2
6	0.01728	0.01734	1225.97	2808.1	2225.6	2755.2	582.4/19.4	52.9/1.8
7	0.01718	0.01724	1226.74	374.3	72.7	406.4	301.6/10.1	32.1/1.1
8	0.01737	0.01743	1225.03	2191.8	1817.3	2185.3	374.5/12.5	6.5/0.2
9	0.02250	0.02254	1157.98	1939.4	2369.5	1888.4	430.0/14.3	51.0/1.7
10	0.01824	0.01830	1212.73	2499.5	2316.5	2409.3	182.9/6.1	90.1/3.0
11	0.01748	0.01754	1223.84	1194.8	973.9	1239.3	221.0/7.4	44.5/1.5
12	0.01743	0.01749	1224.38	2249.5	1837.7	2184.9	411.7/13.7	64.5/2.2
13	0.01905	0.01911	1198.59	2505.7	2506.6	2394.1	0.9/0.0	111.6/3.7
14	0.01873	0.01879	1204.56	967.4	1037.5	1007.7	70.1/2.3	40.3/1.3
15	0.01753	0.01759	1223.24	1656.8	1318.0	1592.1	338.8/11.3	64.7/2.2
16	0.01793	0.01799	1217.89	2937.4	2587.5	2800.6	349.9/11.7	136.8/4.6
17	0.01762	0.01768	1222.36	1647.9	1344.7	1591.6	303.3/10.1	56.3/1.9
18	0.01891	0.01897	1201.49	991.3	1068.9	1007.6	77.7/2.6	16.4/0.5
19	0.01938	0.01943	1193.51	1858.4	1977.0	1811.9	118.6/4.0	46.5/1.6
20	0.01729	0.01735	1225.85	1081.9	810.4	1120.2	271.5/9.0	38.2/1.3
						Average	**281.5/9.38**	**61.8/2.1**

7.4.2 Experiment-II: Performance of Proposed EnLDI Method

In this experiment, the performance of the proposed enhanced leak detection and isolation method is evaluated and compared with traditional fixed model method. To approximate the real working situations, the simulations are implemented under 20 different model parameters, e.g., f the friction coefficient and b the pressure propagation speed. The leak location, pressure head and flow rates are varying at each time. The 20 different leaks are randomly simulated on a 3000m long pipeline. Tab. 7.3 shows the result of model parameter estimation and the comparison of localization accuracy between proposed method and conventional fixed model parameter method.

It can be seen that the varying friction factors f and NPW speed b can be accurately estimated for the 20 leak events by comparing the $2nd$ and $3rd$ columns of Tab. 7.3. By applying the estimated model parameters in leak localization algorithm, the average localization error is at $61.8m$ which is much smaller than the conventional $281.5m$, and the average error percentage decreases from 9.38% to 2.1%. In this experiment, the mean of the 20 true friction factors is taken as the fixed model parameter. It verifies that the proposed EnLDI method which online estimates and updates the model parameters can achieve significant improvement in leak localization accuracy.

7.4.3 Experiment-III: Case Evaluation.

In this experiment, the proposed CDPDF and EnLDI are implemented on the pipeline networks as shown in Fig. 7.6. The pipe parameters are displayed in Tab. 7.1. In this example system, 12 sensors are deployed along the pipeline, the observation duration is $800s$ and the sampling period is $1s$. Therefore, for the whole system, 9600 points of pressure and 9600 points of flow values are acquired. To test the leak detection and isolation performance, 4 consecutive leaks are simulated at different time instants on each of the pipes. The leak instants and locations information can be viewed at the "True" values included in Tab. 7.5 .

The data fusion performance is evaluated under the condition of $\epsilon = 0.1m$ and $\epsilon = 0.1m^3/s$ for the pressure head H and flow rate Q respectively. Without applying the proposed method, for each of the sensor, 800 transmissions of pressure data and 800 transmissions of flow data are demanded. By viewing the result of transmissions in Tab. 7.4, most of the transmissions can be eliminated. For the example of PIPE-I, only 755 times of transmission are needed to acquire 4800 data points. It can decrease more than 80% of the transmissions in the whole system, thus, the life-span of the distributed sensors can be significantly extended.

Fig. 7.11 shows the leak monitoring plots of the 4 pipes. As shown in the left column, the leaks can be effectively detected by viewing the significant

Table 7.4: Transmission times for the sensors of Experiment-III.

PIPE	S_1		S_2		S_3		Transmission out of 4800	Percentage eliminated
	H_1	Q_1	H_2	Q_2	H_3	Q_3		
I	169	93	154	85	168	86	755	**83.24%**
II	232	95	148	78	77	76	706	**84.33%**
III	220	62	182	64	97	75	700	**84.47%**
IV	206	50	234	72	171	73	806	**82.11%**

increase after each leak instant as marked with red vertical lines and the plots converge to the estimated leak size values respectively. In the right column of Fig. 7.11, the estimated leak location plots are displayed. It can be seen that the plots converge to constant values which are the corresponding leak locations. The detected leak time instants and leak locations are listed in Tab. 7.5 . The average leak localization error of the 4 leaks is $61m$ which is about 2.56% of the pipe length.

By viewing the overall performance of proposed methods, all the leaks are effectively detected and isolated regardless of the pipe sizes and the flow conditions, the average localization error of proposed EnLDI is only $61m$ and up to more than 80% of the transmissions can be eliminated by the proposed CD-PDF method. It implies that by employing the proposed methods, the WSNs life-span can be significantly extended and the leak localization accuracy is also improved.

Table 7.5: Leak detection and isolation result of Experiment-III.

PIPE	Leak Instant(s)		Leak Location(m)		Error/%
	True	Detected	True	Estimated	
I	100	115	1900	1791	109/3.63%
II	200	223	600	644	44/2.2%
III	300	329	900	960	60/3%
IV	400	425	1300	1269	31/1.55%
				Average	**61**(m)/**2.56**%

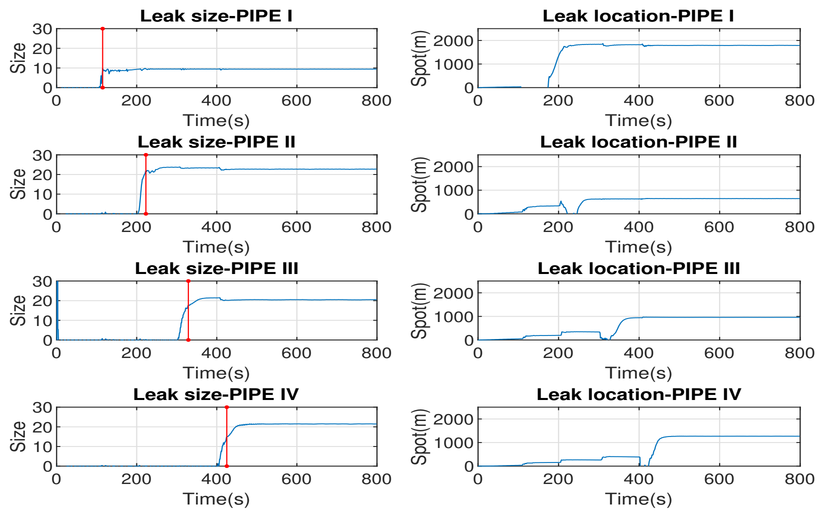

Figure 7.11: Leak detection and isolation plots of Experiment-III.

7.5 Conclusion

In this chapter, two methods, namely CDPDF for data fusion based transmission, and EnLDI for leak diagnosis in WSNs pipeline networks monitoring system are proposed. The CDPDF method can pair with various supervised learning based predictors. Instead of only considering the auto-correlation of individual sensor node, the cross-correlation among neighbored sensors is also considered. The prediction rate is significantly improved and the data transmission amount can be greatly decreased, leading to great save in energy consumption. Furthermore, in the ENLDI method, pipeline model parameters are estimated online which can handle model uncertainties caused by the varying working situations. It improves the pipeline model quality and provides more accurate leak detection and isolation result. The performance of proposed methods is thoroughly evaluated in several case studies performed on a simulated pipeline model according to a real scenario. The results demonstrate that the proposed methods can achieve higher leak detection and isolation accuracy with much less data transmission than existing methods in the WSNs pipeline monitoring system.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

This thesis has explored the problems of applying ML methods in redundant transmission reduction, communication channel equalization and signal detection, and event detection in WSN systems. Among various ML algorithms, the ELM and BiLSTM are mainly applied and discussed. The advantage of fast learning speed and good approximation ability of ELM has been exploited to solve the problems such as data fusion, communication channel equalization, preliminary event detection in WSN. The strong time-series processing ability of BiLSTM has been applied in sequence classification, time-series prediction in WSN monitoring system. The conclusions can be summarized in the following:

- In Chapter 3, to reduce the energy consumed on wireless transmission, a prediction-based data fusion method based on grey model combining OPELM is proposed. The number of wireless transmission can be greatly reduced by accurately predicting the next step measurement. The low computational complexity of ELM enables it to be employed in fast online working situations. It has provided an effective data fusion method for WSN systems which can significantly reduce the energy cost on wireless transmission.

- In Chapter 4, the issue of wireless communication channel distortion between sink node and base station is considered. The multi-ELM regressor based channel equalization and signal detection methods are proposed in OFDM systems. The proposed methods can achieve better detection accuracy, lower computational complexity, more robust activation function and subchannel numbers adaptability compared to other ELM based equalizers.

- In Chapter 5, the event detection in WSNs is investigated in an example of pipeline leak detection. An OPELM combining BiLSTM leak detection method is proposed for pressure wave analysis. Firstly, OPELM is applied to roughly single out the abrupt leak alike pressure plummets, then BiLSTM is applied to further identify the true leaks by exploiting its promising sequential pattern recognition capability. Compared to traditional pressure based leak detection methods, the proposed one can significantly drop the false alarm rate and increase the detection accuracy.

- In Chapter 6, the pipeline leak detection and localization is investigated in multi-sensor scenario. A disturbance assisted method is proposed based on BiLSTM classification. In this proposed method, the traditional ignored non-leak disturbances are identified and exploited for NPW propagation speed estimation. Thus, the propagation speed which is assumed constant by conventional methods can be online updated and the localization accuracy is significantly improved. The proposed method achieves better leak detection and localization accuracy compared to the traditional methods.

- In Chapter 7, the WSN based transmission reduction and pipeline leak monitoring are further studied in the more complex pipeline networks scenario. Multi-sensors are monitoring the pipeline networks with continuous pressure and flow acquisition. An ELM based data fusion method is proposed to reduce the redundant wireless transmissions. The proposed method considers both the temporal and spatial correlations among neighboring sensors to establish the prediction model. Furthermore, a modified model-based pipeline leak detection and localization method is proposed which can online update the model parameters and improve the model approximation. The proposed method can achieve better leak detection and localization accuracy with much less data transmissions compared to traditional method.

8.2 Future Work

In the future research, more DL-based algorithms are to be explored and applied in WSN systems. For example, the RNN based methods are promising in time-series prediction and sequential data classification. However, it usually needs a long learning period to obtain a satisfactory prediction model which jeopardizes their practical online implementation in WSNs. The techniques of extracting more representative features from raw samples may relieve the complexity of the learning process. The feature extraction techniques are to be researched and combined with more deep learning methods to solve some common issues in WSN such as data fusion, target localization and tracking, etc. Furthermore, in fault detection systems, the faulty event related data samples are rare and valuable, however, the normal state data is mostly available. The imbalanced knowledge of data samples is challenging the accuracy of model learning by ML methods. Some deep learning methods such as generative adversarial network can be used to mock or replicate the information from faulty events, thus, improve the model approximation ability. More studies are also to be implemented in this field where the training data samples are imbalanced.

Bibliography

[1] International Electrotechnical Commission, "Internet of Things: Wireless Sensor Networks," *International Electrotechnical Commission(IEC) White Paper*, `http://www.iec.ch/whitepaper/pdf/iecWP-internetofthings-LR-en.pdf` (accessed January 9, 2019).

[2] T. Alhmiedat, A. A. Taleb, and M. Bsoul, "A study on threads detection and tracking systems for military applications using WSNs," *International Journal of Computer Applications*, vol. 40, no. 15, pp. 12–18, 2012.

[3] L. M. L. Oliveira and J. J. P. C. Rodrigues, "Wireless sensor networks: A survey on environmental monitoring," *Journal of Communications*, vol. 6, no. 2, pp. 143–151, 2011.

[4] Y. Zhang, L. Sun, H. Song, and X. Cao, "Ubiquitous WSN for healthcare: Recent advances and future prospects," *IEEE Internet Things Journal*, vol. 1, no. 4, pp. 311–318, 2014.

[5] R. A. Alzafarani and G. A. Alyahya, "Energy efficient IoT home monitoring and automation system," *2018 15th Learning and Technology Conference*, pp. 107–111, 2018.

[6] J. Chinrungrueng, U. Sunantachaikul, and S. Triamlumlerd, "Smart parking: An application of optical wireless sensor network," *2007 International Symposium on Applications and the Internet - Workshops*, pp. 7–10, 2007.

[7] J. Amutha, S. Sharma, and J. Nagar, "WSN strategies based on sensors, deployment, sensing models, coverage and energy efficiency: review, ap-

proaches and open issues," *Wireless Personal Communications*, vol. 111, no. 2, pp. 1089–1115, 2020.

[8] T. Bala, V. Bhatia, S. Kumawat, and V. Jaglan, "A survey: Issues and challenges in wireless sensor network," *International Journal of Engineering and Technology*, vol. 7, no. 2, pp. 53–55, 2018.

[9] S. R. Beeram and S. Kuchibhotla, "A survey on state-of-the-art financial time series prediction models," *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, pp. 596–604, 2021.

[10] Y. Sun, R. Wang, B. Sun, W. Li, and F. Jiang, "Prediction about time series based on updated prediction ARMA model," *Proceedings-2013 10th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2013*, no. 91024008, pp. 680–684, 2013.

[11] L. Zhang and L. L. Yang, "Machine learning for joint channel equalization and signal detection," *Machine Learning for Future Wireless Communications*, John Wiley & Sons, Ltd, 2020, pp. 213–241.

[12] K. Burse, R. N. Yadav and S. C. Shrivastava, "Channel equalization using neural networks: A review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 40, no. 3, pp. 352–357, 2010.

[13] I. Santos, J. J. Murillo-Fuentes, J. C. Aradillas and E. Arias-De-Reyna, "Channel equalization with expectation propagation at smoothing level," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 2740–2747, 2020, doi: 10.1109/TCOMM.2020.2975624.

[14] M. A. Ahad, S. Paiva, G. Tripathi, and N. Feroz, "Enabling technologies and sustainable smart cities," *Sustain. Cities Soc.*, vol. 61, p. 102301, 2020.

[15] E. M. Jovanovska and D. Davcev, “No pollution smart city sightseeing based on WSN monitoring system,” *2020 Sixth International Conference on Mobile and Secure Services (MobiSecServ)*, 2020, pp. 1–6.

[16] A. Khan, S. Gupta, and S. K. Gupta, “Multi-hazard disaster studies: Monitoring, detection, recovery, and management, based on emerging technologies and optimal techniques,” *International Journal of Disaster Risk Reduction*, vol. 47, p. 101642, 2020.

[17] A. J. Ramadhan, “Smart water-quality monitoring system based on enabled real-time internet of things,” *Journal of Engineering Science and Technology*, vol. 15, no. 6, pp. 3514–3527, 2020.

[18] M. Abdelhafidh, M. Fourati, L. C. Fourati, A. Mnaouer and M. Zid, “Novel data preprocessing algorithm for WSN lifetime maximization in water pipeline monitoring system,” *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2019.

[19] T. O. Ayodele, “Introduction to machine learning,” *New Advances in Machine Learning*, InTech, pp. 1–9, 2010.

[20] T. M. Mitchell and others, “Machine learning,” first ed., McGraw-Hill, Inc., New York, NY, USA, 1997.

[21] P. Langley and H. A. Simon, “Applications of machine learning and rule induction,” *Commun. ACM*, vol. 38, no. 11, pp. 54–64, 1995.

[22] D. Praveen Kumar, T. Amgoth, and C. S. R. Annavarapu, “Machine learning algorithms for wireless sensor networks: A survey,” *Information Fusion*, vol. 49, pp. 1–25, 2019.

[23] T. Gao, S. Wu, F. Bu, G. Pang and J. Song, “Source routing protocol based on Bayesian network for wireless sensor networks,” *International Conference on Intelligent Computing, Automation and Systems (ICICAS),* 2019, pp. 773–777.

[24] J. V. Maisuria and S. N. Mehta, "Bayesian-based spectrum sensing and optimal channel estimation for MAC Layer Protocol in cognitive radio sensor networks," *The Computer Journal,* no. 6, pp. 942—957, 2020,

[25] T. Zhang, Q. Zhao, K. Shin and Y. Nakamoto, "Bayesian-optimization-based peak searching algorithm for clustering in wireless sensor networks," *Journal of Sensor and Actuator Networks*, vol. 7(1):2, 2018

[26] L. B. Bhajantri, B. G. Kumbar, "Cluster-based data aggregation in wireless sensor networks: A Bayesian classifier approach," in *Evolutionary Computing and Mobile Sustainable Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol. 53, Springer, Singapore.

[27] T. Wang, X. Wang and W. Shi, et al., "Target localization and tracking based on improved Bayesian enhanced least-squares algorithm in wireless sensor networks," *Computer Networks*, vol. 167, no. 106968, 2020.

[28] P. Qian, Y. Guo, N. Li and S. Yang, "Variational bayesian inference-based multiple target localization in WSNs with quantized received signal strength," *IEEE Access*, vol. 7, pp. 60228-60241, 2019.

[29] R. Mohanraj, J. Xu and E. Babulak "A secure energy efficient IoT based fractional correlated bayesian data transmission in WSNs," *Journal of Communications and Information Networks*, vol. 4, no. 1, pp. 54–66, 2019.

[30] S. Mahmood, I. A. Samad, and C. Hassan, et al., "Standalone noise and anomaly detection in wireless sensor networks: A novel time-series and adaptive Bayesian-network-based approach," *Software: Practice and Experience*, vol. 50, no. 4, pp. 428–446, 2020.

[31] D. K. Sharma, Aayush, A. Sharma and J. Kumar, "KNNR:K-nearest neighbour classification based routing protocol for opportunistic networks," *2017 Tenth International Conference on Contemporary Computing*, pp. 1–6, Noida, India.

[32] A. Juwaied, L. Jackowska-Strumiłło, A. Sierszeń, (2019) Modified TEEN Protocol in Wireless Sensor Network Using KNN Algorithm. In: Choraś M., Choraś R. (eds) Image Processing and Communications Challenges 10. IP&C 2018. Advances in Intelligent Systems and Computing, vol. 892, Springer, Cham.

[33] Y. Liu, Z. Ma and F. Yu, "Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy," *Knowledge-Based Systems*, vol. 133, pp. 208–220, 2017.

[34] Y. Chen, X. Hu and W. Fan, et al., "Fast density peak clustering for large scale data based on kNN," *Knowledge-Based Systems*, vol. 187, no. 104824, 2020.

[35] X. Fang, Z. Jiang, L. Nan and L. Chen, "Optimal weighted K-nearest neighbour algorithm for wireless sensor network fingerprint localisation in noisy environment," *IET Communications*, vol. 12, no. 10, pp. 1171–1177, 2018.

[36] H. Wang, K. Lyu and H. Jiang, et al., "Location algorithm for WSNs with mobile anchor node based on optimized K-Nearest neighbors," *Eleventh International Conference on Advanced Computational Intelligence (ICACI)*, pp. 190–195, June 7–9, 2019, Guilin, China.

[37] Z. Wang, G. Song and C. Gao, "An isolation-based distributed outlier detection framework using nearest neighbor ensembles for wireless sensor networks," *IEEE Access*, vol. 7, pp. 96319-96333, 2019.

[38] F. Khan, S. Memon, and S. H. Jokhio, "Support vector machine based energy aware routing in wireless sensor networks," *2016 2nd International Conference on Robotics and Artificial Intelligence (ICRAI)*, pp. 1–4, Rawalpindi, Pakistan.

[39] S. Kamalesh and P. G. Kumar, "Data aggregation in wireless sensor network using SVM-based failure detection and loss recovery, " *Journal of Ex-*

perimental & Theoretical Artificial Intelligence, vol. 29, iss. 1, pp. 133–147, 2017.

[40] S. Yadav and R. S. Yadav, "Redundancy elimination during data aggregation in wireless sensor networks for IoT systems," *Recent trends in communication, computing, and electronics*, Springer, 2019, pp. 195–205.

[41] F. Zhu and J. Wei, "Localization algorithm for large scale wireless sensor networks based on Fast-SVM," *Wireless Personal Communications*, vol. 95, no. 3, pp. 1859–1875, 2017.

[42] X. Wang, Z. Zhao and T. Wang, et al., "A LS-SVM based measurement points classification algorithm for adjacent targets in WSNs," *Sensors*, vol. 19, iss. 24, no. 5555, 2019.

[43] Z. Zhao, X. Wang and T. Wang, "A novel measurement data classification algorithm based on SVM for tracking closely spaced targets," *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 4, pp. 1089–1100, 2019.

[44] T. K. Dao, T. T. Nguyen and J. S. Pan, et al., "Identification failure data for cluster heads aggregation in WSN based on improving classification of SVM," *IEEE Access*, vol. 8, pp. 61070–61084, 2020.

[45] G. M. Borkar, L. H. Patil, D. Dalgade and A. Hutke, "A novel clustering approach and adaptive SVM classifier for intrusion detection in WSN: A data mining concept," *Sustainable Computing: Informatics and Systems*, vol. 23, pp. 120–135, 2019.

[46] A. Mehmood, Z. Lv, J. Lloret and M. M. Umar, "ELDC: An artificial neural network based energy-efficient and robust routing scheme for pollution monitoring in WSNs," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 106–114, 2020.

[47] A. A. Khan, M. S. Jamal, and S. Siddiqui, "Dynamic duty-cycle control for wireless sensor networks using artificial neural network (ANN)," *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 420–424.

[48] Y. Harold Robinson, E. Golden Julie, S. Balaji, and A. Ayyasamy, "Energy Aware Clustering Scheme in Wireless Sensor Network Using Neuro-Fuzzy Approach," *Wirel. Pers. Commun.*, vol. 95, no. 2, pp. 703–721, 2017.

[49] F. Khorasani and H. R. Naji, "Energy efficient data aggregation in wireless sensor networks using neural networks," *Int. J. Sens. Networks*, vol. 24, no. 1, pp. 26–42, 2017.

[50] B. Madagouda and R. Sumathi, "Artificial neural network approach using mobile agent for localization in wireless sensor networks," *Adv. Sci. Technol. Eng. Syst. J.*, vol. 6, pp. 1137–1144, 2021.

[51] S. R. Jondhale, M. Sharma, R. Maheswar, R. Shubair, and A. Shelke, "Comparison of neural network training functions for RSSI based indoor localization problem in WSN," *Handbook of Wireless Sensor Networks: Issues and Challenges in Current Scenario's*, Springer, 2020, pp. 112–133.

[52] C. Zhou, L. Wang, and Z. Lu, "The study of WSN node localization method based on back propagation neural network," *International Conference on Applications and Techniques in Cyber Security and Intelligence*, pp. 458–466, Edizioni della Normale, Cham, 2017.

[53] P. R. Chandre, P. N. Mahalle, and G. R. Shinde, "Intrusion detection and prevention using artificial neural network in wireless sensor networks," *Proceeding of First Doctoral Symposium on Natural Computing Research: DSNCR 2020*, vol. 169, p. 113, 2021.

[54] S. Mohapatra, P. M. Khilar, and R. R. Swain, "Fault diagnosis in wireless sensor network using clonal selection principle and probabilistic neural network approach," *Int. J. Commun. Syst.*, vol. 32, no. 16, p. e4138, 2019.

[55] L. Liu, J. Xu and J. Shu "A link quality estimation method based on improved weighted extreme learning machine," *IEEE Access*, vol. 9, pp. 11378–11392, 2021.

[56] L. Zhao, W. Zhao, A. Hawbani, et al., "Novel online sequential learning-based adaptive routing for edge software-defined vehicular networks," *IEEE Transactions on Wireless Communications*, vol. 20, no. 5, pp. 2991–3004, 2021.

[57] I. Ullah and H. Y. Youn, "Efficient data aggregation with node clustering and extreme learning machine for WSN," *The Journal of Supercomputing*, vol. 76, pp. 10009–10035, 2020.

[58] L. Cao, Y. Cai and Y. Yue, et al., "A Novel Data Fusion Strategy Based on Extreme Learning Machine Optimized by Bat Algorithm for Mobile Heterogeneous Wireless Sensor Networks," *IEEE Access*, vol. 8, pp. 16057–16072, 2020.

[59] Z. E. Khatab, A. Hajihoseini and S. A. Ghorashi, "A fingerprint method for indoor localization using autoencoder based Deep Extreme Learning Machine," *IEEE Sensors Letters*, vol. 2, iss. 1, pp. 1–4, 2018.

[60] S. Phoemphon, C. So-In and D. Niyato, "A hybrid model using fuzzy logic and an extreme learning machine with vector particle swarm optimization for wireless sensor network localization," *Applied soft computing*, vol. 65, pp. 101–120, 2018.

[61] W. J. Zhang, D. Z. Han, K. C. Li and F. I. Massetto, "Wireless sensor network intrusion detection system based on MK-ELM," *Soft Computing*, vol. 24, pp. 12361–12374, 2020.

[62] S. Duraisamy, G. K. Pugalendhi and P. Balaji, "Reducing energy consumption of wireless sensor networks using rules and extreme learning machine algorithm", *The Journal of Engineering*, vol. 2019, iss. 9, pp. 5443–5448, 2019.

[63] A. M. George and S. Y. Kulkarni, "Cluster based routing protocols for IOT application," *Int. J. Comput. Netw. Inf. Secur.*, vol. 5, pp. 43—49, 2019.

[64] H. K. Hoomod and T. K. Jebur, "Applying self-organizing map and modified radial based neural network for clustering and routing optimal path in wireless network," *Journal of Physics: Conference Series*, 2018, vol. 1003, no. 1, p. 12040.

[65] M. H. Shafiabadi, A. K. Ghafi, D. D. Manshady, and N. Nouri, "New method to improve energy savings in wireless sensor networks by using SOM neural network," *J. Serv. Sci. Res.*, vol. 11, no. 1, pp. 1-–16, 2019.

[66] U. M. Kulkarni, H. H. Kenchannavar, and U. P. Kulkarni, "Self-organising map-based dynamic decision-making algorithm for heterogeneous wireless sensor network," *Int. J. Parallel, Emergent Distrib. Syst.*, vol. 36, no. 4, pp. 312-–334, 2021.

[67] T. Kitanouma, E. Nii, N. Adachi and Y. Takizawa, "SmartFinder: Cloud-based self organizing localization for mobile smart devices in large-scale indoor facility," *2017 Global Internet of Things Summit (GIoTS)*, 2017, pp. 1–6.

[68] E. Kfoury, J. Saab, P. Younes, and R. Achkar, "A self organizing map intrusion detection system for rpl protocol attacks," *Int. J. Interdiscip. Telecommun. Netw.*, vol. 11, no. 1, pp. 30-–43, 2019.

[69] K. Huang, Q. Zhang, C. Zhou, N. Xiong, and Y. Qin, "An efficient intrusion detection approach for visual sensor networks based on traffic pattern learning," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 10, pp. 2704-–2713, 2017.

[70] M. Behzad, M. S. Javaid, M. A. Parahca, and S. Khan, "Distributed PCA and consensus based energy efficient routing protocol for WSNs," *J. Inf. Sci. Eng.*, vol. 33, no. 5, pp. 1267–1283, 2017.

[71] P. H. Li and H. Y. Youn, “Distributed stochastic principal component analysis using stabilized Barzilai-Borwein step-size for data compression with WSN,” *J. Supercomput.*, pp. 1-–20, 2021.

[72] X. Zhang, H. Wu, Q. Li and B. Pan, “An event-based data aggregation scheme using PCA and SVR for WSNs,” *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pp. 1–5.

[73] Z. Jellali, L. N. Atallah and S. Cherif, “Principal component analysis based clustering approach for WSN with locally uniformly correlated data,” *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 174–179.

[74] S. S. Banihashemian and F. Adibnia, “A new range-free PCA-based localization algorithm in wireless sensor networks,” *Int. J. Commun. Syst.*, vol. 33, no. 6, p. e4291, 2020.

[75] H. Xu, “Semi-supervised manifold learning based on polynomial mapping for localization in wireless sensor networks,” *Signal Processing*, vol. 172, p. 107570, 2020.

[76] B. Xu, X. Zhang, and L. Liu, “The failure detection method of WSN based on PCA-BDA and fuzzy neural network,” *Wirel. Pers. Commun.*, vol. 102, no. 2, pp. 1657-–1667, 2018.

[77] Q. Sun, Y. Sun, X. Liu, Y. Xie, and X. Chen, “Study on fault diagnosis algorithm in WSN nodes based on RPCA model and SVDD for multi-class classification,” *Cluster Comput.*, vol. 22, no. 3, pp. 6043-–6057, 2019.

[78] R. Madiha, D. D. Ningombam, and S. Shin. “Energy efficient K-means clustering-based routing protocol for WSN using optimal packet size.” *2018 International Conference on Information Networking (ICOIN)*. IEEE, 2018.

[79] B. Barekatain, S. Dehghani, and M. Pourzaferani, “An energy-aware routing protocol for wireless sensor networks based on new combination of ge-

netic algorithm & K-means," *Procedia Comput. Sci.*, vol. 72, pp. 552–560, 2015.

[80] S. K. Mydhili, S. Periyanayagi, S. Baskar, P. M. Shakeel and P. R. Hariharan , "Machine learning based multi scale parallel K-means++ clustering for cloud assisted internet of things," *Peer-to-Peer Networking and Applications*, vol. 13 pp. 2023–2035, 2020.

[81] A. K. Idrees, W. L. Al-Yaseen, M. A. Taam and O. Zahwe, "Distributed data aggregation based modified K-means technique for energy conservation in periodic wireless sensor networks," *2018 IEEE Middle East and North Africa Communications Conference (MENACOMM)*, 2018, pp. 1–6.

[82] S. E. Khediri et al., "Improved node localization using K-means clustering for wireless sensor networks," *Comput. Sci. Rev.*, vol. 37, p. 100284, 2020.

[83] G. Han, H. Wang, M. Guizani, S. Chan and W. Zhang, "KCLP: A k-Means cluster-based location privacy protection scheme in WSNs for IoT," *IEEE Wireless Communications*, vol. 25, no. 6, pp. 84–90, 2018.

[84] D. R. Kumar and A. J. C. Sunder, "Minkowski K-means intrusion detection for mitigating sinkhole and blackhole attack in healthcare wireless sensor network," vol. 14, pp. 1121-–1146, 2018.

[85] Z. Wei, F. Liu, Y. Zhang and J. Xu, et al., "A Q-learning algorithm for task scheduling based on improved SVM in wireless sensor networks," *Computer Networks*, vol. 161, pp. 138–149, 2019.

[86] C. Savaglio, P. Pace and G. Aloi, et al., , "Lightweight Reinforcement Learning for Energy Efficient Communications in Wireless Sensor Networks," *IEEE Access*, vol. 7 pp. 29355–29364, 2019.

[87] K. S. Babu, "Grid clustering and fuzzy reinforcement-learning based energy-efficient data aggregation scheme for distributed WSN," *IET Commun.*, vol. 14, no. 16, pp. 2840–2848(8), 2020.

[88] S. Otoum, B. Kantarci and H. Mouftah, “Empowering reinforcement learning on big sensed data for intrusion detection,” *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–7.

[89] S. Sujanthi and S. N. Kalyani, “SecDL: QoS-aware secure deep learning approach for dynamic cluster-based routing in WSN assisted IoT,” *Wirel. Pers. Commun.*, vol. 114, no. 3, pp. 2135—2169, 2020.

[90] R. Huang et al., “Resilient routing mechanism for wireless sensor networks with deep learning link reliability prediction,” *IEEE Access*, vol. 8, pp. 64857–64872, 2020.

[91] S. N. Mohanty, et al., “Deep learning with LSTM based distributed data mining model for energy efficient wireless sensor networks,” *Phys. Commun.*, vol. 40, p. 101097, 2020.

[92] L. Zhao, “Data aggregation in WSN based on deep self-encoder,” *Int. J. Performability Eng.*, vol. 14, no. 11, pp. 2723-–2730, 2018.

[93] Z. Shen, T. Zhang, A. Tagami, and J. Jin, “When RSSI encounters deep learning: An area localization scheme for pervasive sensing systems,” *J. Netw. Comput. Appl.*, vol. 173, p. 102852, 2021.

[94] S. H. Javadi, A. Guerrero, and A. M. Mouazen, “Source localization in resource-constrained sensor networks based on deep learning,” *Neural Comput. Appl.*, vol. 33, no. 9, pp. 4217—4228, 2021.

[95] S. Otoum, B. Kantarci and H. T. Mouftah, “On the feasibility of deep learning in sensor network intrusion detection,” *IEEE Networking Letters*, vol. 1, no. 2, pp. 68–71, 2019.

[96] O. A. Alzubi, “A deep learning-based frechet and dirichlet model for intrusion detection in IWSN,” *J. Intell. Fuzzy Syst.*, vol. Preprint, pp. 1—11, 2021.

[97] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing,* vol. 70, pp. 489–501, 2006.

[98] G. B. Huang, H. Zhou, X. Ding and R. Zhang, "Extreme learning machine for regression and multiclassclassification", *IEEE Trans. Syst. Man Cybern. PartB: Cybern*, 2012, (42):513-529.

[99] W. Deng,Q. Zheng and L. Chen, "Regularized extreme learning machine," in:*IEEE Symposium on Computational Intelligence and Data Mining*, 2009, pp.389-395.

[100] J. A. K. Suykens, J. De Brabanter, L. Lukas, J. Vandewalle, "Weighted Least Squares Support Vector Machines: Robustness and Sparse Approximation," *Neurocomputing*, 2002, 48:85-105.

[101] K. Zhang and M. Luo, "Outlier-robust Extreme Learning Machine for Regression Problems," *Neurocomputing*, 2015, 151:1519-1527.

[102] P. Bartlett, "The Sample Complexity of Pattern Classification with Neural Networks: the size of the weights is more important than the size of the network," *IEEE Trans. Inf. Theory*, 1998, 44:525-536.

[103] A. K. Hoerl and R. W.Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, 1970, 12(1): 55-67.

[104] K. Chen, Q. Lv and Y. Lu, et.al., "Robust regularized extreme learning machine for regression using iteratively reweighted least squares," *Neurocomputing*, 2017, 230:345-358.

[105] M. Sundermeyer, H. Ney and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517–529, March 2015.

[106] Özal Yildirim, "A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification," *Computers in Biology and Medicine*, vol. 96, pp. 189–202, May 2018.

[107] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Transactoins on Neural Networks and Learning Systems*, vol. 28, no. 10, Oct. 2017.

[108] M. Kocakulak and I. Butun, "An overview of wireless sensor networks towards Internet of Things," *IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, Jan. 2017.

[109] X. Ning, C. G. Cassandras, "Dynamic sleep time control in wireless sensor networks,"*ACM Transactions on Sensor Networks*, vol. 6, No. 3, June 2010.

[110] R. I. Tandel, "Leach protocol in wireless sensor network: A survey," *International Journal of Computer Science and Information Technologies*, vol. 7, No. 4, pp. 1894–1896, 2016.

[111] J. Deng, "Introduction to grey system theory," *The Journal of Grey System*, vol. 1, iss. 1, 1989.

[112] G. Wei, Y. Ling, B. Guo, B. Xiao and A. V. Vasilakos, "Prediction-based data aggregation in wireless sensor networks: Combining grey model and Kalman Filter," *Computer Communications*, vol. 34 iss. 6, pp. 793–802, 2011.

[113] S. Shamshirband, D. Petkovic, H. Javidnia and A. Gani, "Sensor data fusion by support vector regression methodology — A comparative study," *IEEE Sensors Journal*, vol. 15, No. 2, pp. 850–854, 2015.

[114] N. Dawar, S. Ostadabbas and N. Kehtarnavaz, "Data augmentation in deep learning-based fusion of depth and inertial sensing for action recognition," *IEEE Sensors Letters*, vol. 3, No. 1, 2019.

[115] X. Luo, D. Zhang, L. T. Yang, J. Liu, X. Chang, H. Ning, "A kernel machine-based secure data sensing and fusion scheme in wireless sensor networks for the cyber-physical systems," *Future Generation Computer Systems*, vol. 61, pp. 81–96, 2016.

[116] X. Luo and X. Chang, "A novel data fusion scheme using grey model and extreme learning machine in wireless sensor networks," *International Journal of Control, Automation, and Systems*, vol. 13, pp. 539–546, 2015.

[117] E. Kayacan, B. Ulutas and O. Kaynak, "Grey system theory-based models in time series prediction," *Expert Systems with Applications,* vol. 37, pp. 1784–1789, 2010.

[118] H. J. Rong, Y. S. Ong, A. W. Tan, and Z. Zhu, "A fast pruned-extreme learning machine for classification problem," *Neurocomputing,* vol. 72, pp. 359–366, 2008.

[119] Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten and A. Lendasse, "OP-ELM: Optimally pruned extreme learning machine," *IEEE Transaction on Neural Networks,* vol. 21, No. 1, pp. 158–162, 2010.

[120] T. Similä and J. Tikka, "Multiresponse sparse regression with application to multidimensional scaling," *Proc. Int. Conf. Artif. Neural Netw.,* vol. 3697, pp. 97–102, 2005.

[121] R. Myers, Classical and Modern Regression With Applications, 2nd ed. Pacific Grove, CA: Duxbury, 1990.

[122] G. Bontempi, M. Birattari, H. Bersini(1998) "Recursive lazy learning for modeling and control". In: Nédellec C., Rouveirol C. (eds) Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol. 1398. Springer, Berlin, Heidelberg.

[123] G. Charalabopoulos, P. Stavroulakis and A. H. Aghvami1,"A frequency-

domain neural network equalizer for OFDM," *GLOBECOM*, 2003, pp. 571-575.

[124] D. J. Sebald and J. A. Bucklew, "Support vector machine techniques for nonlinear equalization," *IEEE Transactions on Signal Processing*, 2000, 48(11): 3217-3226.

[125] S. Mhatli, H. Mrabet, I. Dayoub and E. Giacoumidis, "A novel support vector machine robust model based electrical equaliser for coherent optical orthogonal frequency division multiplexing systems," *IET Communications*, 2017, 11(7): 1091-1096.

[126] M. Li, G. Huang and P. Saratchandran, et.al., "Fully complex extreme learning machine," *Neurocomputing*, 2005, 68: 306-314.

[127] I. G. Muhammad, K. E. Tepe and E. Abdel-Raheem, "QAM equalization and symbol detection in OFDM systems using extreme learning machine," *Neural Computing and Applications*, 2013, 22: 491-500.

[128] J. Liu, K. Mei, X. Zhang, D. Ma and J. Wei, "Online extreme learning machine-based channel estimation and equalization for OFDM systems," *IEEE Communications Letters*, July 2019, 23(7):1276-1279.

[129] J. Conca, *Pick Your Poison For Crude--Pipeline, Rail, Truck Or Boat.* Forbes, April 26, 2014. [Online]. Available: https://www.forbes.com/sites/jamesconca/2014/04/26/pick-your-poison-for-crude-pipeline-rail-truck-or-boat/#6b7f23a717ac

[130] A. Abdulshaheed, F. Mustapha, A. Ghavamian, "A pressure-based method for monitoring leaks in a pipe distribution system: A review," *Renewable and Sustainable Energy Reviews*, vol.69, pp. 902–911, March 2017.

[131] U. Baroudi, A. A. AL-Roubaiey and A. Devendiran, "Pipeline leak detection systems and data fusion: a survey," *IEEE Access*, vol. 7, pp. 97426–97439, July 2019.

[132] X. Diao, G. Shen, J. Jiang, Q. Chen, Z. Wang, L. Ni, A. Mebarki and Z. Dou, "Leak detection and location in liquid pipelines by analyzing the first transient pressure wave with unsteady friction," *Journal of Loss Prevention in the Process Industries*, vol.60, pp.303–310, July 2019.

[133] I. Santos-Ruiz, J. R. Bermúdez, F. R. López-Estrada, V. Puig, L. Torres and J. A. Delgado-Aguiñaga, "Online leak diagnosis in pipelines using an EKF-based and steady-state mixed approach," *Control Engineering Practice*, vol.81, pp.55–64, Dec. 2018.

[134] Q. C. Jiang, S. F. Yan, H. Cheng and X. F. Yan, "Local-global modeling and distributed computing framework for nonlinear plant-wide process monitoring with industrial big data," *IEEE Transaction on Neural Networks and Learning Systems*, [Online]. Doi: 10.1109/TNNLS.2020.2985223.

[135] Q. C. Jiang, X. F. Yan and B. Huang, "Review and perspectives of data-driven distributed monitoring for industrial plant-wide process," *Industrial and Engineering Chemistry Research*, vol. 58, no. 29, pp. 12899–12912, July, 2019.

[136] M. B. Abdulla, R. O. Herzallah and M. A. Hammad, "Pipeline leak detection using artificial neural network: experimental study," *Proceedings of International Conference on Modeling, Identification and Control (ICMIC)*, pp. 328–332, Cairo, Egypt, Aug. 2013.

[137] R .B. Santos, M. Rupp, S. J. Bonzi and A. M. F. Fileti, "Comparison between multilayer feedforward neural networks and a radial basis function network to detect and locate leaks in pipelines transporting gas," *Chemical engineering transactions*, vol. 32, pp. 1375–1380, 2013.

[138] H. Chen, H. Ye, C. Lv and H. Su, "Application of support vector machine learning to leak detection and localization in pipelines," *Instrument and Measurement Technology Conference*, Como, Italy, May 2004.

[139] R. Xiao, Q. Hu and J. Li, "Leak detection of gas pipelines using acoustic signals based on wavelet transform and Support Vector Machine," *Measurement*, vol. 146, pp. 479–489, June. 2019.

[140] S. Rashida, U. Akram and S. A. Khan, "WML: wireless sensor network based machine learning for leakage detection and size estimation," *Procedia Computer Science*, vol. 63, pp. 171–176, Sep. 2015.

[141] A. Rojik, Endroyono and A. N. Irfansyah, "Water pipe leak detection using the k-nearest neighbor method," in *International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pp. 393–398, Aug. 2019.

[142] H. Zhang, Q. Li, X. Zhang and W. Ba, "Industrial oil pipeline leakage detection based on extreme learning machine method," in *Advances in Neural Networks-ISNN 2017, Part II*, F.Cong et al (Eds.), Sapporo, Hakodate, and Muroran, Hokkaido, Japan: Springer, 2017, pp. 380–387.

[143] A. E. U. Salam, M. Tola, M. Selintung and F. Maricar, "Application of SVM and ELM methods to predict location and magnitude leakage of pipelines on water distribution network," *International Journal of Advanced Computer Research*, vol. 5, no. 19, pp. 140–144, June 2015.

[144] A. Soldevila, R. M. Fernandez-Canti, J. Blesa, S. Tornil-Sin and V. Puig, "Leak detection in water distribution networks using Bayesian classifiers," *Journal of Process Control,* vol. 55, pp. 1–9, July 2017.

[145] Z. Chen, X. Xu, X. Du, J. Zhang and M. Yu, "Leak detection in pipelines using decision tree and multi-support vector machine," in *Proceedings of the 2017 2nd International Conference on Electrical, Control and Automation Engineering (ECAE 2017),* Xiamen, China, December 2017.

[146] T. R. Sheltami, A. Bala and E. M. Shakshuki, "Wireless sensor networks for leak detection in pipelines: a survey," *Journal of Ambient Intelligence and Humanized Computing,* vol. 7 no. 3, pp. 347–356, June 2016.

[147] J. Reynolds and A. Kam, "An evaluation of negative pressure wave leak detection: challenges, limitations, and use cases," *Pipeline Simulation Interest Group Annual Meeting,* London, UK, May 2019.

[148] C. Ma, S. Yu and J. Huo, "Negative pressure wave-flow testing gas pipeline leak based on wavelet transform," in *IEEE International Conference on Computer, Mechatronics, Control and Electronic Engineering(CMCE),* vol. 5, pp. 306–308, 2010.

[149] Z. Peng, J. Wang and X. Han, "A study o negative pressure wave method based on Haar wavelet transform in ship piping leakage detection system," in *IEEE 2nd International Conference on Computing, Control and industrial Engineering(CCIE),* vol. 2, pp. 306–308, 2011.

[150] C. H. Tian, J. C. Yan, J. Huang, Y. Wang, D. S. Kim and T.Yi, "Negative pressure wave based pipeline leak detection: challenges and algorithms," in *IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*, Suzhou, July 2012.

[151] J. Liu, D. Zang, C. Liu, Y. Ma and M. Fu, "A leak detection method for oil pipeline based on markov feature and two-stage decision scheme," *Measurement*, vol. 138, pp. 111–113, Jan. 2019.

[152] A. Abdulshaheed, F. Mustapha, A. Ghavamian, "A pressure-based method for monitoring leaks in a pipe distribution system: A review," *Renew. Sust. Energ. Rev.*, vol.69, pp. 902–911, March 2017.

[153] D. Zaman *et al.*, "A review of leakage detection strategies for pressurised pipeline in steady-state," *Eng. Fail. Anal.*, vol. 109, no. 104264, 2020.

[154] P. Xu, R. Du and Z. Zhang, "Predicting pipeline leakage in petrochemical system through GAN and LSTM," *Knowl.-Based Syst.,* vol. 175, pp. 50–61, 2019.

[155] L. Yang and Q. Zhao, "A novel PPA method for fluid pipeline leak detection based on OPELM and bidirectional LSTM," in *IEEE Access*, vol. 8, pp. 107185–107199, 2020.

[156] C. Liu *et al.*, "New leak-localization approaches for gas pipelines using acoustic waves," *Measurement*, vol. 134, pp. 56–65, Feb. 2019.

[157] J. Wan, Y. Yu, Y. Wu, R. Feng and N. Yu, "Hierarchical leak detection and localization method in natural gas pipeline monitoring sensor networks," *Sensors*, vol. 12, no. 1, pp. 189–214, Jan. 2012.

[158] X. Lang, J. Cao, and P. Li, "Localization method of multiple leaks based on time-frequency analysis and improved differential evolution," *IEEE Sensor Journal*, vol. 20,no. 23, pp. 14383–14390, 2020.

[159] X. Lang, "Leak localization method for pipeline based on fusion signal," *IEEE Sensor Journal*, vol. 21, no. 3, pp. 3271–3277, 2021.

[160] T. B. Quy and J. M. Kim, "Crack detection and localization in a fluid pipeline based on acoustic emission signals," *Mech. Syst. Signal Process.*, vol. 150, p. 107254, 2021.

[161] M. R. Delgado and O. B. Mendoza, "A comparison between leak location methods based on the negative pressure wave," *14th Int. Conf. Electr. Eng. Comput. Sci. Autom. Control*, Mexico, Sep. 2017.

[162] M. T. Ismail and A. A. A. Dghais, "A comparative study between discrete wavelet transform and maximal overlap discrete wavelet transform for testing stationarity," *International Journal of Mathematical Science and Engineering*, vol. 7, no. 12, 2013.

[163] M. Müller, "Dynamic time warping," in *Information Retrieval for Music and Motion*, Berlin, Heidelberg, Springer, 2007, ch. 4, pp. 69–84.

[164] E. N. Mosland, K. D. Lohne,B. Ystad and A. Hallanger, “Pressure wave velocity in fluid-filled pipes with and without deposits in the low-frequency range,” *J. Hydraul. Eng.*, vol. 144, no. 10, Oct. 2018.

[165] M. Basseville, “On fault detectability and isolability,” *European Journal of Control*, vol. 7, no. 6, pp. 625–637, 2001.

[166] Z. Abbas, M. R. Anjum, M. U. Younus, and B. S. Chowdhry, “Monitoring of Gas distribution pipelines network using wireless sensor networks,” *Wireless Personal Communications*, vol. 117, pp. 2575–2597, April 2021, doi:10.1007/s11277-020-07997-6.

[167] A. Moubayed, M. Sharif, M. Luccini, S. Primak, A. Shami, “Water leak detection survey: challenges & research opportunities using data fusion & federated learning,” *IEEE Access*, vol. 9, pp. 40595–40611, March 2021, doi:10.1109/ACCESS.2021.3064445.

[168] M. Y. Aalsalem, W. Z. Khan, W. Gharibi, M. K. Khan and Q. Arshad, “Wireless sensor networks in oil and gas industry: Recent advances, taxonomy, requirements, and open challenges,” *Journal of Network and Computer Applications*, vol. 113, pp. 87–97, July 2018, doi: 10.1016/j.jnca.2018.04.004.

[169] T. R. Sheltami, A. Bala and E. M. Shakshuki, “Wireless sensor networks for leak detection in pipelines: a survey,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, pp. 347–356, June 2016, doi: 10.1007/s12652-016-0362-7.

[170] T. Meng, X. Jing, Z. Yan and W. Pedrycz, “A survey on machine learning for data fusion”, *Information Fusion*, vol. 57, pp. 115–129, May 2020, doi: 10.1016/j.inffus.2019.12.001.

[171] E. F. Nakamura, A. A. F. Loureiro and A. C. Frery, “Information fusion for wireless sensor networks: Methods, models, and classifications”, *ACM Computing Surveys*, Sep. 2007, doi: 10.1145/1267070.1267073.

[172] G. Wei, Y. Ling, B. Guo, B. Xiao and A. V. Vasilakos, "Prediction-based data aggregation in wireless sensor networks: Combining grey model and Kalman Filter", *Computer Communications*, 34(2011), 793–802, doi: 10.1016/j.comcom.2010.10.003.

[173] M. Ashouri, H. Yousefi, J. Basiri, A. M. A. Hemmatyar, "PDC: Prediction-based data-aware clustering in wireless sensor networks", *Journal of Parallel and Distributed Computing*, vol. 81–82, July 2015, pp. 24–35, doi: 10.1016/j.jpdc.2015.02.004.

[174] L. Yang, Q. Zhao and Y. Jing, "Conditional training based GM and GM-OPELM data fusion schemes in wireless sensor networks," *2019 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*, Victoria, BC, Canada, 21-23 Aug. 2019, doi: 10.1109/PACRIM47961.2019.8985084.

[175] A. Ayadi, Q. Ghorbel, M. S. BenSalah and M. Abid, "Spatio-temporal correlations for damages identification and localization in water pipeline systems based on WSNs," *Computer Networks*, vol. 171, article 107134, April 2020, doi: 10.1016/j.comnet.2020.107134.

[176] X. Song, Y. Liu, L. Xue, J. Wang, J. Zhang, J. Wang, L. Jiang and Z. Cheng, "Time-series well performance prediction based on Long Short-Term Memory (LSTM) neural network model," *Journal of Petroleum Science and Engineering*, vol. 186, 100682, Mar. 2020, doi:10.1016/j.petrol.2019.106682.

[177] Z. Shen, Y. Zhang, J. Lu, J. Xu and G. Xiao, "A novel time series forecasting model with deep learning," *Neurocomputing*, vol. 396, pp. 302–313, July 2020, doi:10.1016/j.neucom.2018.12.084.

[178] A. Sagheer, M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks", *Neurocomputing*, vol. 323, Jan. 2019, pp. 203–213, doi: 0.1016/j.neucom.2018.09.082.

[179] S. S. Namini, N. Tavakoli, A. S. Namin, "The performance of LSTM and BiLSTM in forecasting time series", *2019 IEEE International Conference on Big Data*, Los Angeles, CA, USA, 9-12 Dec. 2019, doi: 10.1109/BigData47090.2019.9005997.

[180] S. Panigrahi, R. M. Pattanayak, P. K. Sethy and S. K. Behera, "Forecasting of sunspot time series using a hybridization of ARIMA, ETS and SVM methods", *Solar Physics*, vol. 296(6), 2021, doi:10.1007/s11207-020-01757-2.

[181] T. T. Chen, S. J. Lee, "A weighted LS-SVM based learning system for time series forecasting", *Information Sciences*, vol. 299, Apr. 2015, pp. 99–116, doi: 10.1016/j.ins.2014.12.031.

[182] J. A. Jaramillo and J. D. Velásquez, "Research in financial time series forecasting with SVM: contributions from literature", *IEEE Latin America Transactions*, vol. 15, Iss. 1 Jan. 2017, pp. 145–153, doi: 10.1109/TLA.2017.7827918.

[183] M. Larrea, A. Porto, E. Irigoyen, A. J. Barragán, J. M. Andújar, "Extreme learning machine ensemble model for time series forecasting boosted by PSO: Application to an electric consumption problem," *Neurocomputing*, vol. 396, pp. 305–313, July. 2020, doi:10.1016/j.neucom.2018.12.084.

[184] L. Xiao, W. Shao, F. Jin and Z. Wu, "A self-adaptive kernel extreme learning machine for short-term wind speed forecasting," *Applied Soft Computing*, vol. 99, 106917, Feb. 2021, doi:10.1016/j.asoc.2020.106917.

[185] B. Yadav, S. Ch, S. Mathur and J. Adamowski, "Discharge forecasting using an Online Sequential Extreme Learning Machine (OS-ELM) model: A case study in Neckar River, Germany", *Measurement*, vol. 92, Oct. 2016, pp. 433–445, doi: 10.1016/j.measurement.2016.06.042.

[186] Z. M. Yaseen, M. F. Allawi, A. A. Yousif, O. Jaafar, F. M. Hamzah and A. E. Shafie, "Non-tuned machine learning approach for hydrological time

series forecasting", *Neural Computing and Applications*, vol. 30, Sep. 2018, pp. 1479–1491, doi: 10.1007/s00521-016-2763-0.

[187] S. Razvarz., R. Jafari, A. Gegov, "Leakage modelling for pipeline" in *Flow Modelling and Control in Pipeline Systems. Studies in Systems, Decision and Control.* Cham, Switzerland: Springer, 2021, doi: 10.1007/978-3-030-59246-2_6

[188] M. H. Chaudhry, *Applied Hydraulic Transients*, 3rd ed., 2014, doi: 10.1007/978-1-4614-8538-4.

[189] N. Visairo, C. Verde, "Leak detection conditions in a pipeline via a geometric approach," *IFAC Proceedings Volumes*, vol. 36(5), June 2003, pp. 921–926, doi: 10.1016/S1474-6670(17)36610-7.

[190] C. Verde, "Accommodation of multi-leak location in a pipeline," *Control Engineering Practice*, vol. 13(8), Aug. 2005, pp. 1071–1078, doi: 10.1016/j.conengprac.2004.09.010.

[191] M. Guillén and J. F. Dulhoste, G. Besançon, I. Rubio S., R. Santos and D. Georges, "Leak detection and location based on improved pipe model and nonlinear observer," *2014 European Control Conference (ECC)*, Strasbourg, France, 24-27 June. 2014, doi: 10.1109/ECC.2014.6862388.

[192] P. Liu, S. Li and Z. Wang, "Multi-leak diagnosis and isolation in oil pipelines based on Unscented Kalman filter," *2018 Chinese Control And Decision Conference (CCDC)*, Shengyang, China, 9-11 June. 2018, doi: 10.1109/CCDC.2018.8407496.

[193] Y. He, S. Li and Y. Zheng, "Distributed state estimation for leak detection in water supply networks," *IEEE/CAA Journal of Automatica Sinica(Early access)*, Jan. 2017, pp. 1–9, doi: 10.1109/JAS.2017.7510367.

[194] X. Liu, "A survey on clustering routing protocols in wireless sensor networks," *Sensors*, vol. 12, pp. 11113–11153, 2012.