Pipeline Monitoring: Leveraging Attention-based 1DCNN-BiLSTM for Accurate Leak Detection with Minimal False Alarms

***Abstract*— Pipelines are an essential infrastructure for the transportation of fluids and gases in many industries. Leaks in pipelines, however, present a significant concern because of their potential environmental and economic impact. Both economic and environmental losses can be prevented if leaks are detected in a timely and accurate manner. In recent studies, deep learning models, particularly sequential models, have demonstrated promising capabilities in anomaly detection for time series data. However, detecting leaks while keeping false alarms to a minimum remains a challenge for leak detection systems. Unfortunately, leak detection systems fail to detect leaks while keeping false alarms to a minimum. This paper presents a novel leak detection approach that offers improved results over base models while keeping the number of false positives low at the industrial level. We introduced the CB-AttentionNet model based on a 1D convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM), and multi-head attention mechanisms that can capture local and long time series dependencies. Experimental results show that the detection performance of our method is better both in terms of accuracy and minimizing false negatives in different simulated scenarios in pipeline industries.**

***Index Terms*— Leak Detection, CNN, BiLSTM, Attention mechanisms, Pipeline, Deep Learning**

I. Introduction

PIPELINES have become an integral part of infrastructure in North America. Providing reliable and expansive energy distribution is crucial for sustaining economic activities and meeting the daily energy needs of the region. Accordingly, in the United States, 3.3 million miles of regulated pipelines form a critical infrastructure network, and 64% of U.S. energy commodities are efficiently transported through these pipelines [1]. Despite their undeniable utility, potential leaks in pipelines pose a significant concern. Leaks can have consequences ranging from environmental disasters such as fire outbreaks to the severe endangerment of human lives. The Deepwater Horizon oil spill, the largest marine oil spill in history, occurred in 2010 when the Deepwater Horizon oil rig exploded in the Gulf of Mexico and subsequently sank two days later. The devastating impact of this spill resulted in the deaths of an estimated 800,000 birds, with around 1,400 whales and dolphins found stranded by the end of 2015. The spill also had a significant toll on local bird species, with a devastating loss of approximately 12 percent of brown pelicans and over 30 percent of laughing gulls in the affected area [2].

Pipeline leaks can be caused by a variety of factors, including internal corrosion resulting from chemical interactions between transported substances and pipeline materials over time, which poses a persistent threat to the integrity of the system. Physical damage, stemming from external forces like excavation work, accidental impacts, and natural disasters, can also compromise the structural soundness of pipelines. Additionally, mistakes in operations, such as insufficient maintenance, mishandling, or procedural errors, can increase the chances of leaks [3]. These leaks may exhibit various patterns, making them resemble other abnormal behaviors, which can be challenging to differentiate. Consequently, systems may struggle to accurately detect leaks that are similar to other anomalies such as side activities near pipelines, leading to either poor judgment or an increase in false alarms. False alarms can disrupt normal operations and lead to unnecessary shutdowns or interventions. Therefore, addressing leaks accurately poses several challenges that require a delicate balance between leak detection accuracy and simultaneously reducing false alarms [4].

The study of identifying leaks in pipelines encompasses a diverse range of methodologies, falling primarily into the two main categories of hardware-based and software-based. Hardware-based methods involve external systems and physical devices to detect and monitor pipeline leaks [5], [3]. A wide range of sensor technologies, including acoustic, pressure, vibration, mobile wireless, fiber optic, infrared cameras, radar-based, and electrokinetic methods, are employed for leak detection in piplines. However, the challenges of maintaining extensive sensor infrastructure make software-based models increasingly favored for cost-effective and practical pipeline monitoring over long distances [6]. Software-based methods focus on internal systems and algorithms that process the collected data to detect and pinpoint the exact location of leaks [7]. Machine learning algorithms offer a continuous monitoring solution by analyzing data from hardware-based sensors, distinguishing between normal variations and potential leaks, and thus enhancing leak detection systems' reliability. Over the past few decades, several deep learning models have been applied to detect time-series leaks. These models find local and global dependencies to find patterns in the data. For example, the work of [8] introduced a novel architecture that combined a 3D CNN with a long short-term memory (LSTM) network to capture both spatial and temporal information in video frames. The model could learn complex patterns associated with methane leaks in pipelines. Similarly, authors of [9] focused on the application of DNN and random forest classifiers for source tracking of chemical leaks using fence monitoring data. They introduced a model called Gated Residual Networks (GRes-Nets), which combines Deep Residual Networks (ResNets) with a gate mechanism. Also, As can be seen in [10], variations of CNNs were used on spectrograms to improve performance. For example, AlexNet, VGG-16, and ResNet-18 were used to train leak signals. A CNN model adapted from a pretrained AlexNet network was leveraged to demonstrate the proposed approach using vibration data collected from an experimental pipeline test bed. The CNN model was found to be able to detect leakages on polyvinyl chloride (PVC) pipelines.

Nevertheless, the majority of prior studies fail to take into account false alarms while addressing leaks. The effectiveness of leak detection is undoubtedly one of the most important factors, but it is equally important to recognize anomalies that resemble leaks but are false positives.

Unlike previous leak detection research, we propose a CB-AttentionNet model based on convolutional neural networks (CNNs), short-term memory (BiLSTMs), and multihead attention mechanisms. Our contributions in this paper are as follows: 1) To the best of our knowledge, we proposed a novel CB-AttentionNet model that can detect leaks in pipelines at the industrial level. This model can capture local and long dependencies while assigning different weights to different parts of the input sequence, allowing the model to give more importance to certain time steps during the learning process. This is particularly useful for identifying relevant patterns in the dataset. 2) Validation of our model using industrial data. Comparative analysis with previous detection methods demonstrates the superior robustness of our model in accurately detecting true positives while minimizing false positives. 3) Introduction of a post-smoothing method aimed at addressing the increased false positive rate, especially in cases where there is a similarity between leaks and other anomalies. This method contributes to the refinement of leak detection outcomes. The rest of the paper is organized as follows: In Section II, we discuss the related works on leak detection. Section III contains the methodology proposed, and in Sections IV and V, we present experimental results, our dataset, and conclusions.

II. Literature Review

This section reviews machine learning and deep learning approaches in leak detection in petroleum pipelines presented in the literature. The existing challenges and various solutions are discussed and the distinctions of our approach are highlighted. In addition, we presented applications of the combined CNN and BiLSTM models in other domains.

A. Machine learning approaches for leak detection

In order to increase safety and prevent fatalities, environmental disasters, and economic losses, it is highly desirable to develop automated pipeline monitoring that utilizes artificial intelligence (AI) and machine learning (ML) [11]. For example, in [12], the research team established an experimental configuration by deploying a network of sensors within a water distribution system to gather real-time information regarding water flow and pressure. The collected data were pre-processed and utilized as input for an ML model designed for leak detection. The authors employed decision trees and random forest algorithms to classify the data and achieve accurate leak detection.

The work in [13] used a hybrid approach that combines SVM, K-nearest neighbor (KNN), Gaussian mixture model (GMM), and Naive Bayes in multi-dimensional feature space. Due to the large amount of acoustic data in pipelines, traditional algorithms often have trouble handling leaks. These algorithms can become computationally expensive and inefficient as the volume of data grows. A novel approach is introduced in [14], called tnGAN (Trinetworks-based Generative Adversarial Network) to address leak detection issues in incomplete sensor data. The tnGAN includes a generative model to recover missing data by leveraging the similarity of data features at the same level. It incorporates a multiview awareness strategy to enhance temporal and spatial characteristics and introduces a dual-discriminative network architecture to detect pipeline status by comparing latent features of samples.

In another work, [15] was introduced enhanced real-time automated gas leak detection using a hybrid LSTM and attention mechanism (AM) network. By assigning initial weights with a fully-connected AM neural network, the approach achieves superior accuracy compared to state-of-the-art methods to find leaks and their locations.

In the work of [16], a magnetic anomaly detection method using deep learning neural networks (DLNN) was introduced to enhance accuracy in locating buried pipelines. The DLNN, optimized for improved performance, achieves nearly %90 prediction accuracy. In another work [17], they combined deep learning with transient frequency response (TFR) for water pipeline leak detection. The aforementioned work employs extended TFR equations to generate a deep learning set, using a pre-calibrated hydraulic model to extract frequency response functions (FRFs). The fully linear DenseNet (FL-DenseNet) is trained on simulated FRFs, exhibiting robustness against uncertainties and surpassing accuracy compared to methods relying on inaccurate models for identifying leaks in water pipelines. While previous methodologies yielded accurate results, their diverse input datasets, originating from various formats of pipeline signals, hinder meaningful comparisons. We established three baselines (LSTM AM, LSTM AE, and Fully Linear DenseNet (FL-DenseNet)) based on prior research and applied them to our dataset. Our experiments demonstrate superior performance not only in accuracy, recall, and precision but also in minimizing the false positive rate. An elevated false positive rate poses a significant risk to industrial operations, potentially leading to adverse consequences and increased expenditures. This is why minimization is important [18].

B. Application of similar methodologies

The use of 1D Convolutional Neural Networks (CNNs) for capturing specific features within time series data has been extensively explored in academic literature. A significant early contribution to this concept comes from the work of [19]. Although [19] focuses primarily on generating raw audio waveforms, it introduces the use of dilated causal convolutions in a deep 1D CNN architecture, which can be used to analyze sequential data like time series. Subsequently, this innovation has influenced the adoption of 1D CNNs across various domains to extract local patterns and features from time series data.

However, it is not always easy to find patterns due to their length and complexity; therefore, other methodologies have been developed to cope with this issue. Early on, according to the work of [20], despite the paper's primary focus on image captioning and the use of LSTMs for sequence generation, it shows how convolutional and recurrent networks can be combined to understand images. This paper proposes a CNN-LSTM model to predict gas field production in a southwest China gas reservoir. By leveraging CNN's feature extraction and LSTM's sequence dependence learning, the model captures changing production trends. Introducing a novel strategy (PURPS), it estimates some input features based on predicted gas and water production. Results demonstrate the model's effectiveness, outperforming existing methods with lower MAPE errors. Another application of combining recurrence networks and convolutions is this paper [21]. This study introduces a novel approach using PPG signals and a CNN-LSTM model for categorizing blood pressure (BP) levels. This approach addresses the need for continuous BP monitoring to detect and prevent health issues early. The Transformer's innovative architecture enabled more parallelization during training, resulting in faster computations and longer sequences. Additionally, the self-attention mechanism enhanced the model's ability to comprehend context and relationships by capturing global dependencies in input sequences. One of the applications of this method in time-series data is presented in [22]. This study addresses the challenge of accurate time series forecasting by proposing a hybrid model integrating long short-term memory (LSTM) and multi-head attention, aiming to capture complex non-linear patterns. The model's performance is compared with standard forecasting techniques across 16 datasets, demonstrating superior results in terms of symmetric mean absolute percentage error (SMAPE) and achieving the best average rank (AR) among the methods tested. In a separate study [23] employing a framework that combines long short-term memory (LSTM) and attention mechanisms to forecast machine attitude and position, the LSTM-attention model exhibits superior accuracy compared to the LSTM model. This improvement is evident in the increased mean value of the coefficient of determination (R2) and the decreased mean value of the root mean square error (RMSE).

III. Methodology

This section introduces a leak detection integrated model (CB-AttentionNet) outlined in Figure 1. In addition to 1DCNN, BiLSTM, and an attention mechanism, this model consists of three distinct components. Using these components, the CB-AttentionNet captures sequential short and long dependencies within leak detection datasets. Algorithm **??** shows the framework of our proposed model. The algorithm proceeds by iterating over the dataset in windows of the specified size. Within each window, it applies a series of computational steps to discern leaks within a dataset. The explanation for each step will be provided later on in this section.

```
Algorithm 1: Framework for the proposed model
Input  : Window of size $w$ data, number of iterations $i$
Output: Flag indicating if it's a leak
Procedure:;
foreach window do
    $feature\_maps \leftarrow$ Apply 1D-CNN to window;
    $hidden\_layers \leftarrow$ Apply BiLSTM on
      $feature\_maps$;
    foreach attention_head in heads do
        Apply attention_head on $hidden\_layers$;
    $mean\_attention \leftarrow$ Compute mean from the
      attentions;
    for $i \leftarrow 1$ to $i$ do
        Apply iteration $i$ and store result in buffer;
$final\_result \leftarrow$ Compute result based on the buffer;
if $final\_result$ meets criteria then
    return Leak Detected;
else
    return No Leak;
```

A. One dimensional Convolutional Neural Network

An example of a CNN [24] is a type of neural network that can detect simple patterns as well as complex patterns in data, and is frequently used for computer vision. For learning the features of an image, these models usually accept two-dimensional input images with color channels. The layers include convolutional layers for extracting features from input data, pooling layers for reducing spatial dimensions, and fully connected layers for making predictions [24].

A 1D CNN, a variation of the Convolutional Neural Network tailored for one-dimensional sequences, is commonly employed for feature extraction from sequential data like time series or structured tabular data. It excels at detecting local patterns within these sequences by utilizing convolutional operations along the temporal or sequential dimension [24]. The convolutional operations performed by 1D CNNs involve sliding small filters or kernels across the input sequence, enabling the network to focus on localized regions at various positions. As a result of this process, the network can detect and emphasize patterns within subsets of the sequence, enabling it to recognize distinctive local features and structures .

First, our model performs a 1D convolutional operation, where a filter slides over the sequence of timestamps t and performs element-wise multiplications to find local patterns.

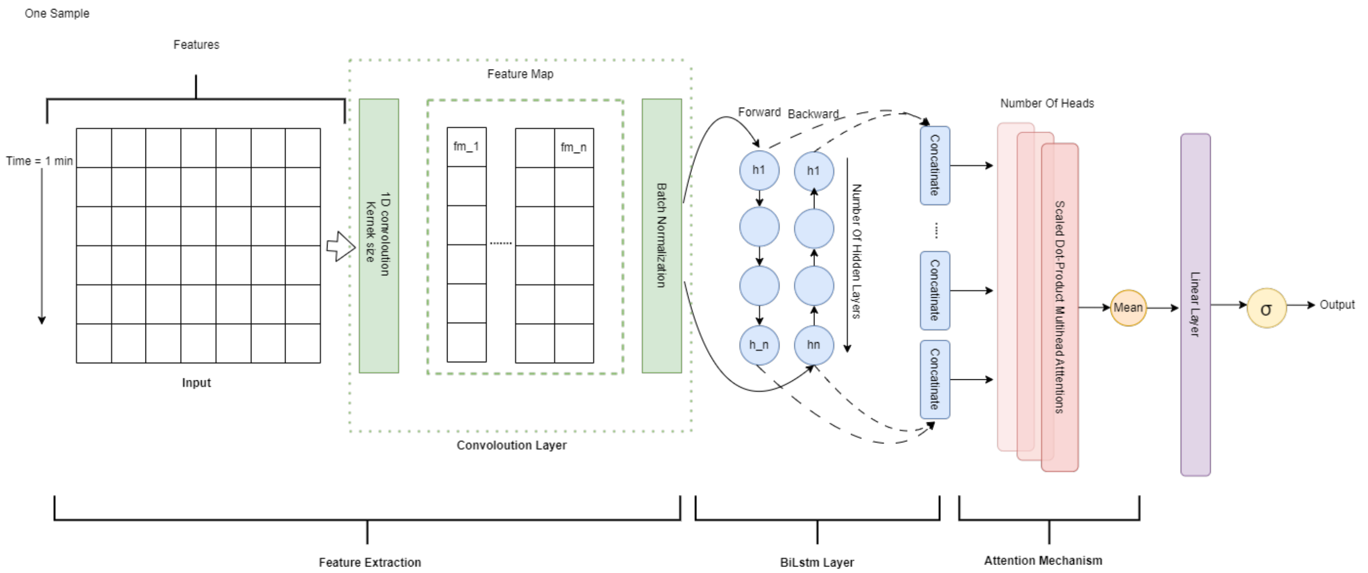

Fig. 1. CB-Attentionnet framework for leak detection in pipelines. The illustrated framework is for each window with a size $\boldsymbol{w}$. The framework consists of 1DCNN for extracting feature maps, a BiLSTM layer and multihead attentions.

To operate convolution of position t, we used a kernel size of K, on a window of feature X_t:

$$Y_t = f\left(\sum_{k=0}^{K-1} W_k \cdot X_{t+k} + b\right) \tag{1}$$

where:

- Y_t: Output for the sequence of timestamps t.
- $f()$: Activation function.
- W_k: Weight at kernel position k.
- X_{t+k}: Input value at position $t+k$.
- b: Bias term.

Y_t value is repeated across all input sequences, producing a series of feature maps that represent patterns extracted from the input data. During the forward propagation of CNN, high-level representations of patterns are extracted from the input time series data. These feature maps were then applied as input data to the second part of our model to process hierarchical dependencies in the data.

B. Bidirectional Long Short-Term Memory

The second segment of our model features the BiLSTM, as it shows, a specialized recurrent neural network (RNN) [25] variant known for its bidirectional nature, effectively capturing information from past and future contexts within sequential data. This BiLSTM architecture includes LSTM units with memory cells, input gates, forget gates, and output gates. It can handle long-distance dependencies and address challenges like vanishing or exploding gradients that are common in RNNs. The LSTM operates by sequentially updating its internal state, considering the current input and the previous hidden state. However, by using a BiLSTM, the input sequence is processed in both forward and backward directions. This structure allows for a more comprehensive understanding of the sequential data, enabling it to capture relative information from both earlier and later states, thus enhancing its ability to analyze the data. Our network structure for the RNN module follows a BiLSTM layer and Dropout layers on feature maps. While a 1D CNN is proficient at capturing local patterns and extracting relevant features from the data, it might not fully grasp long-range dependencies or nuanced relationships existing across a sequence. In our sequential dataset, understanding both short-term patterns and long-term dependencies is crucial for accurate analysis and prediction. Comprehensive monitoring of leaks across different temporal scales can be achieved by integrating both short and long dependencies, which ensures a more effective and reliable detection mechanism. The BiLSTM architecture consists of two LSTM networks: one processing the input sequence in the forward direction, and the other processing it in the backward direction. We applied these layers to the result of the previous part.

BiLSTM output = BiLSTM(previous feature maps as x_t).

If h_t represents the forward and backward hidden states at time step t, combining both forward and backward weights in each hidden state formula for x_t is as follows:

$$\text{BiLSTM:} \quad (\overrightarrow{\mathbf{h}}_t, \overleftarrow{\mathbf{h}}_t) = \text{LSTM}(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}, \overleftarrow{\mathbf{h}}_{t+1}) \tag{2}$$

Then the final output is the concatenation of both backward and forward operations.

$$\mathbf{h}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t] \tag{3}$$

For each hidden state, the LSTM operation for calculating weight is as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{ix}\mathbf{x}_t + \mathbf{W}_{ih}\mathbf{h}_{t-1} + \mathbf{b}_i), \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{fx}\mathbf{x}_t + \mathbf{W}_{fh}\mathbf{h}_{t-1} + \mathbf{b}_f), \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{ox}\mathbf{x}_t + \mathbf{W}_{oh}\mathbf{h}_{t-1} + \mathbf{b}_o), \\
\mathbf{g}_t &= \tanh(\mathbf{W}_{gx}\mathbf{x}_t + \mathbf{W}_{gh}\mathbf{h}_{t-1} + \mathbf{b}_g), \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned} \tag{4}$$

i_t controls the flow of new information into the cell state by applying a sigmoid activation to the weighted sum of the input, previous hidden state, and bias vector. The forget gate f_t modulates what information to discard from the cell state based on the input, previous hidden state, and bias vector through a sigmoid activation. The output gate. o_t regulates the information passed to the next hidden state using a sigmoid activation applied to a similar weighted sum. The cell input g_t integrates new information into the cell state using the hyperbolic tangent function. The cell state. c_t is updated by a combination of the forget gate's action on the previous cell state and the input gate's effect on the cell input.

C. Attention Mechanism

While the 1DCNN-BiLSTM model captures local patterns (via CNN) and long-term dependencies (via BiLSTM), it might struggle with capturing precise and essential information from the entire sequence. When analyzing long sequences as input, it is important to recognize that not all patterns are equally important. CNN-BiLSTM model may consider all inputs equally, including noise and less relevant information. As a result of calculating the impact of input on output, the attention mechanism determines how much weight it should give to the current input.

The attention mechanism [26] works by assigning different weights to the hidden states outputted by the BiLSTM layer, emphasizing certain parts of the sequence when making predictions. First, the model calculates the relevance scores, or attention scores, for each hidden state. In this formula, H represents hidden layers from the previous model, and W_a and H_a are weight matrix.

$$\text{Attention}(\mathbf{H}) = \text{softmax}(\mathbf{W}_a \tanh(\mathbf{U}_a \mathbf{H}^\top + \mathbf{b}_a))\mathbf{H} \quad (5)$$

We apply a softmax function to obtain attention weights that indicate the importance of each hidden state. Then it computes the context vector by combining the hidden states with their respective attention weights. As a result of this process, the model is able to focus more on relevant parts of the input sequence by assigning higher weights (through attention scores) to hidden states that are more important for generating context at each time step. There are four attention mechanisms in our structure. The process described above is performed multiple times in parallel, generating different sets of attention-weighted representations (outputs). Each set is produced by a separate attention head with a different learning weight matrix. Then the output from each attention head is concatenated and passed through a linear transformation.

D. training parameters

a) input data:

The training data used for the model consists of raw acoustic signals recorded over time from pipelines, encompassing various features. This dataset is carefully balanced to ensure a representative distribution of samples that differs from the generated signals. We believe that this balanced composition enhances the credibility and robustness of our model's results. To effectively process this data, a sliding-time window technique is applied, dividing the dataset into multiple equal-length subsequences denoted as $W = \{w1, w2, ...w_n\}$. Each individual sample within these windows is analyzed independently to determine whether there is a leak. Thus, within each window w, there exists a sequence of variables at time t $\{t_i, i = 1, 2, .., n\}$, aiding in the identification of leaks. Breaking down the data into manageable subsequences reduces computational complexity and enhances real-time leak detection capabilities, allowing for efficient analysis and rapid detection of leaks.

b) output data:

If a leak occurs, the model outputs a flag. The crucial aspect about leaks is their consistent timing. Various environmental factors around pipelines can trigger false alarms, resembling the patterns of leaks. When an irregular activity occurs that looks similar to a leak, it's vital to remember that leaks have a predictable and constant pattern over time. Therefore, the model's output is stored in a buffer, and the following sequence of results, denoted as i_1,i_2, .. i_m is examined. If there's a leak in these m subsequent occurrences, the model raises a flag. But if there isn't, it can be other abnormal activities.

IV. Experimental Results

To evaluate the performance of our model, CB-AttentionNet, we used an industrial time series dataset and compared the results of our model with baselines from previous methodologies explained in the literature review section.

A. Dataset Description

Fiber optics can serve as a continuous linear sensor for pipelines, installed along their length to detect various types of dynamic energy irregularities: acoustics, strain, and temperature [27]. This setup allows us to gather real-time data on these variables over time. Our dataset comprises industrial real-time series data that includes simulated instances of leaks and other abnormal activities, providing valuable information for analyzing the model's performance in real-world scenarios. This dataset is structured in three dimensions, incorporating timestamps and features like temperature, strain, and the pipeline's area. As mentioned earlier, our model was trained on smaller subsequences to yield real-time outcomes. For training, we used 80% of the dataset, including totals of more than 20 hours, encompassing non-leak signals, actual leaks, and other abnormal activities. The remaining 20% constituted our test data, which comprises different scenarios used to evaluate the model's false positive rates, recall, precision, and F1 measure for performance comparison.

B. Evaluation Metrics

In evaluating our leak detection model, we employed a set of metrics to assess its performance across different scenarios.

For the test dataset containing leaks, we focused on precision, recall, F1 score, false negative rate, and accuracy. By measuring precision and recall, we were able to judge whether the model was capable of making accurate positive predictions and catching actual positives. The F1 score provided a balanced measure considering both false positives and false negatives. The false negative rate specifically highlighted the proportion of actual leaks incorrectly identified as non-leaks. Meanwhile, for the test dataset featuring abnormal behaviors other than leaks, we concentrated on the false positive rate. Comparing this rate with similar rates from base models enabled us to understand how effectively our model differentiated between different types of abnormal behavior.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{6}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{7}$$

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

$$\text{False Positive Rate} = \frac{\text{False Positives}}{\text{Actual Positives}} \tag{9}$$

C. Preprocessing

In dealing with acoustic signals or any data that exhibits varying levels of noise or intensity, the issue of disparate scales among features often arises. This discrepancy can heavily influence machine learning models, as certain features with higher magnitudes might dominate the modeling process, leading to biased results. This challenge requires normalization and scaling techniques. In order to standardize the data across various areas and features, means and standard deviations for z-score were separately calculated for each feature within each area using exclusively non-leak data. This procedure aimed to create a consistent scale for each feature in every area, ensuring fair comparisons by dividing each data point by its respective standard deviation. This standardized the data distributions to have a mean of 0 and a standard deviation of 1, enabling equitable analysis across different features and areas while reducing the influence of non-leak data variations.

D. Training Model

To train the model, three distinct segments were employed. Initially, the process began with a 1D-CNN utilizing convolution layers with a 3 * 3 kernel size to extract vertical feature maps from each variable. Following this, a BiLSTM configuration was implemented, comprising three layers with a hidden size of 64. Additionally, a dropout layer with a 0.2 probability was integrated to prevent overfitting. Subsequently, there were four attention heads, followed by 0.2 probability dropout layers. The last 2 layers are a linear year with non linear layer (sigmoid function). Throughout training, an Adam optimizer with a learning rate of 0.0002 was utilized. The learning scheduler was parameterized with a step size of 7 and a gamma value of 0.1 to optimize the learning process and mitigate overfitting challenges. Both training and evaluation were implemented by Pytorch version 2.1.1 and CUDA 11.8. We conducted training on our dataset for 50 epochs, implementing early stopping when the accuracy of training data starts to degrade. At the 30th epoch, our model achieved an accuracy of 89.9%, coupled with a loss value of 0.313 showed in figure 2.

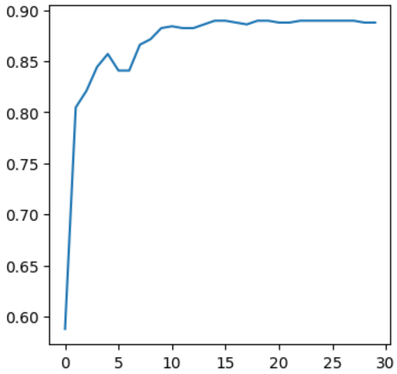

Fig. 2. training accuracy over 30 epochs. The Y axis represents accuracy, while the X axis represents epochs.

E. Postprocessing

To differentiate leaks from other anomalies that share similarities, a post-processing technique was employed. Leaks, known for their continuous activity, persist until interrupted. We stored outputs and, upon detection of a potential leak, awaited confirmation in the following three iterations. If these subsequent iterations also flagged a potential leak, the initial iteration's output was similarly flagged. This method utilized the consistent behavior of leaks across consecutive iterations to confidently identify and flag outputs linked to uninterrupted, continuous activity, aiding in their distinction from other types of anomalies. Figures 5 and 7 show an example of a false positive rate before and after postprocessing. The test dataset as a heatmap is a side activity near a pipeline simulation.

F. Results

The results of our model were obtained after experiments based on the process described. We based our results on two different scenarios. For all simulations, we conducted 5-fold cross-validation on each dataset, and the reported results represent the average performance across these folds. As part of the first scenario, leaks occurred in two specific pipelines on different dates over a period of time.

In the first simulation of that, we utilized our model to examine signal data within one-minute intervals on Windows. The results show that among a total of 23 minutes observed and 10 minutes of leaks, all leaks were detected with an accuracy of 95.83%, precision of 99%, and recall of 89%, respectively. There is only one occurrence of false negatives. During that single minute window when a leak occurred (highlighted in green in figure 3), and our model detected the leak with a delay of less than a minute. Because our system maintains results in a buffer for three iterations as part of its process, this delay did not affect its overall performance.

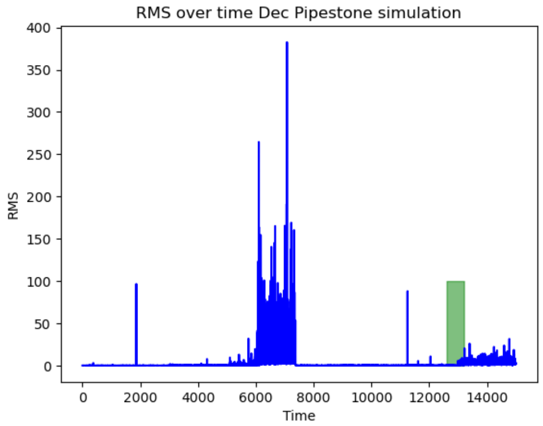

Fig. 3. RMS of Amplitude is shown across the timeline per millisecond in first simulation. An irregular spike or pulse corresponds to leaks, and a highlighted area indicates a leak that was detected late. Not-highlighted areas are true positives.

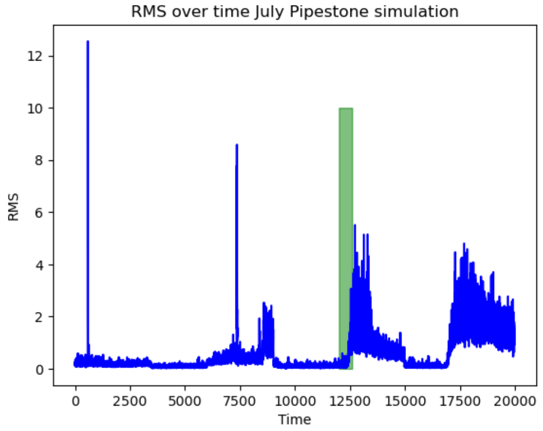

Fig. 4. RMS of Amplitude is shown across the timeline per millisecond in second simulation. An irregular spike or pulse corresponds to leaks, and a highlighted area indicates a leak that was detected late. Not-highlighted areas are true positives.

The second simulation, similar to the first one shown in figure 4 has a window size of one minute. It was tested on 24 minutes of observed signals with a total of 10 minutes of leaks. Accuracy, precision, and recall are 96%, 99%, and 93%. In the same way as before, highlighted areas have delayed leak detection. There are no false-positives in either case.

In the second phase of our test, we evaluated how effectively our model performed when confronted with different anomalies. Specifically, we conducted tests on two distinct side activities near pipelines to assess the occurrence of false positives. Each dataset is approximately 2 hours of simulation. Figures 5 and 6 show the timeline distribution of false positives before post-processing was implemented. The initial accuracy rates are 99.52%, resulting in a total of 200 false alarms for one dataset, and 99.82%, yielding only 5 false alarms for the other. However, following the application of post-processing techniques, the number of false alarms was reduced to 3 and zero for each dataset, respectively. Figures 7 illustrate the updated counts of false positives after this post-processing was applied in first data, showcasing the notable decrease in false alarms for both datasets.

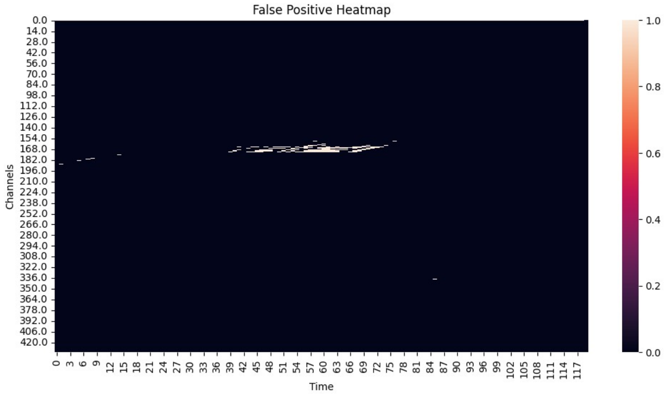

Fig. 5. False positive heatmap in different areas of pipelines (known as channels) over time (one minute intervals) for the first side activity dataset. This plot shows false positives that are detected in each area over time.

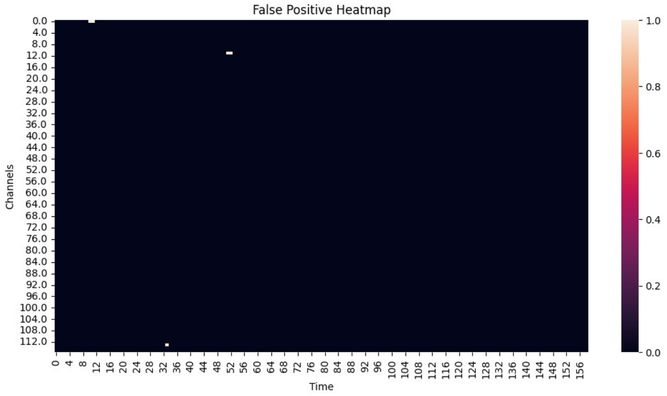

Fig. 6. False positive heatmap in different areas of pipelines (known as channels) over time (one minute intervals) for the second side activity dataset. This plot shows false positives that are detected in each area over time.

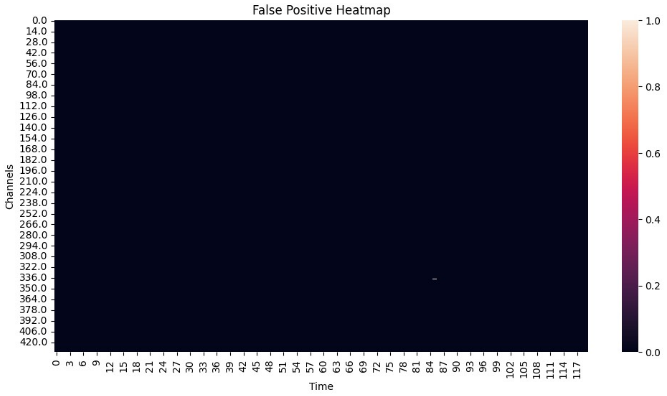

Fig. 7. False positive heatmap for different areas of pipelines over time for the first side activity dataset after postprocessing.

Table I shows the false positive rate for the second simulation before and after processing.

We conducted a comparative analysis between our model and two established leak detection models that had previously demonstrated efficacy in leak detection. As a baseline, we also considered our model without attention mechanisms (1DCNN-

BiLSTM). To ensure a fair comparison, all models were trained on our dataset. Table II showcases the performance metrics—Accuracy, Precision, Recall, and F1 Score—of these models. Subsequently, in evaluating our second simulations involving side activities, we specifically compared the False Positive Rates (FPRs) among these models in table III. Our primary objective was not only to detect leaks effectively but also to minimize the occurrence of false alarms.

TABLE I
COMPARISON OF FALSE POSITIVE RATE BEFORE AND AFTER POST-PROCESSING

	False positive Rates	
	First	Second
Before post-Processing	0.476%	0.002%
After post-Processing	0.005%	0%

The wide range of signal-capturing technologies makes evaluating leak detections challenging so in order to evaluate our model's effectiveness, we compared it with other methodologies that have previously proven effective. All models were trained on our dataset, segmented into one-minute data windows. Using our model with attention mechanism when compared with the CNN-LSTM model demonstrated a significant reduction in false alarm rate, dropping from 0.04 to 0.006 when compared to the base model. Though both models achieved nearly identical accuracy, the substantial difference in false positives indicates our model's ability to effectively minimize false alarms, indicating its potential superiority when it comes to leak detection. FL-DenseNet is a variant on DenseNet. We set the growth rate and the number of layers within each dense block (12 and 4) in order, and we set the number of kernels to 3 in accordance with our model. The LSTM AE and LSTM AM models both have 4 layers and 64 hidden layers. Clearly Attention part of LSTM AM model effected on the number false negatives compare to LSTM AE. However, Compared to the other models, the performs of our model overall is better both for precision and recall, and false positive rates as shown in table II and III.

TABLE II
COMPARISON OF PERFORMANCE METRICS FOR THREE DIFFERENT MODELS IN SIMULATIONS 1 AND 2

		Performance Metrics			
Simulations	Model	Accuracy	Precision	Recall	F1 Score
Simulation 1	CB-Attention	**0.958**	**0.99**	0.89	**0.936**
	LSTM AE	**0.90**	**0.91**	**0.91**	0.89
	FL-DenseNet	0.86	0.86	0.85	0.855
	CNN-LSTM	**0.91**	0.86	**0.90**	0.88
	LSTM AM	**0.94**	**0.92**	**0.96**	**0.94**
Simulation 2	CB-Attention	**0.96**	**0.999**	**0.937**	**0.961**
	LSTM AE	**0.90**	0.88	**0.90**	0.89
	FL-DenseNet	0.88	0.86	0.88	0.87
	CNN-LSTM	**0.94**	**0.94**	0.87	**0.904**
	LSTM AM	**0.96**	**0.95**	**0.96**	**0.95**

TABLE III
COMPARISON OF FPR FOR THREE DIFFERENT MODELS IN SIMULATIONS 1 AND 2

	CB-Attention	LSTM AE	LSTM AM	FL-DenseNet	DLNN
FPR	0.002%	0.12%	0.13	0.08%	0.04%

V. CONCLUSION AND FUTURE WORK

This paper introduces a novel approach, the CB-AttentionNet model, integrating a 1D convolutional neural network (CNN), bidirectional long short-term memory (BiLSTM) to address environmental and economic repercussions caused by the potential for leaks. We conducted two simulations to evaluate the efficacy of our proposed method. Employing our CB-AttentionNet model, we analyzed signal data at one-minute intervals on Windows. The results from this simulation are notable; our model achieved an accuracy of 95.83%, a precision of 99%, and a recall of 89% in first simulation. In the second phase of our simulations, we tested our model's performance in detecting different anomalies near pipelines, aimed at assessing false positive occurrences. Initial accuracy rates were 99.52% (resulting in 200 false alarms) and 99.82% (resulting in only five false alarms). Implementing post-processing techniques notably reduced false alarms to 3 and zero for each dataset, respectively. Assessing the efficacy of a leak detection system involves considering numerous factors that can impact its performance. One crucial aspect is the window size used for segmenting sequential data, which may vary across different use cases. In our forthcoming research endeavors, we aim to analyze the influence of varying time series sequential lengths on leak detection using our proposed model to evaluate the accuracy and capabilities of the leak detection system, contributing valuable insights to optimize the system for diverse applications of anomalies.

REFERENCES

[1] Pipeline and Hazardous Materials Safety Administration. "Title of the webpage." U.S. Department of Transportation. (Year), [Online]. Available: https://www.phmsa.dot.gov/.

[2] Britannica, *Deepwater Horizon oil spill*, https://www.britannica.com/event/Deepwater-Horizon-oil-spill/Environmental-costs, [Accessed July 4, 2023], 2023.

[3] S. Li, Y. Song, and G. Zhou, "Leak detection of water distribution pipeline subject to failure of socket joint based on acoustic emission and pattern recognition," *Measurement*, vol. 115, pp. 39–44, 2018, ISSN: 0263-2241. DOI: https://doi.org/10.1016/j.measurement.2017.10.021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263224117306498.

[4] X. Diao, G. Shen, J. Jiang, *et al.*, "Leak detection and location in liquid pipelines by analyzing the first transient pressure wave with unsteady friction," in *Proceedings of the IEEE International Conference on Automation Science and Engineering*, Aug. 2022, pp. 123–128. DOI: 10.1109/COASE51646.2022.9601792.

[5] S. Fatemi Aghda, K. GanjaliPour, and K. Nabiollahi, "Assessing the accuracy of tdr-based water leak detection system," *Results in Physics*, vol. 8, pp. 939–948, 2018, ISSN: 2211-3797. DOI: https://doi.org/10.1016/j.rinp.2018.01.027. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2211379717318004.

[6] N. Ullah, Z. Ahmed, and J.-M. Kim, "Pipeline leakage detection using acoustic emission and machine learning algorithms," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2743–2753, Apr. 2021. DOI: 10.1109/TII.2020.3026414.

[7] A. A. A. Lah, R. A. Dziyauddin, and N. M. Yusoff, "Localization techniques for water pipeline leakages: A review," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, no. 2, pp. 1–15, Feb. 2021. DOI: 10.1109/TIM.2020.3035064.

[8] "Videogasnet: Deep learning for natural gas methane leak classification using an infrared camera," *Energy*, vol. 238, p. 121 516, 2022, ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2021.121516. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544221017643.

[9] J. Cho, H. Kim, A. L. Gebreselassie, and D. Shin, "Deep neural network and random forest classifier for source tracking of chemical leaks using fence monitoring data," *Journal of Loss Prevention in the Process Industries*, vol. 56, pp. 548–558, 2018, ISSN: 0950-4230. DOI: https://doi.org/10.1016/j.jlp.2018.01.011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950423017304254.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: 10.1109/5.726791.

[11] D. Kampelopoulos, G. N. Papastavrou, G. P. Kousiopoulos, N. Karagiorgos, S. K. Goudos, and S. Nikolaidis, "Machine learning model comparison for leak detection in noisy industrial pipelines," in *2020 9th International Conference on Modern Circuits and Systems Technologies, MOCAST 2020*, Sep. 2020. DOI: 10.1109/MOCAST49295.2020.9200261.

[12] J. A. Coelho and A. Glória, "Precise water leak detection using machine learning and real-time sensor data," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2444–2454, Apr. 2020. DOI: 10.1109/TII.2019.2943237.

[13] S. Rashid, U. Akram, and S. A. Khan, "Wml: Wireless sensor network based machine learning for leakage detection and size estimation," *Procedia Comput Sci*, vol. 63, pp. 171–176, Jan. 2015. DOI: 10.1016/J.PROCS.2015.08.329.

[14] X. Li, V. Metsis, H. Wang, and A. H. H. Ngu, "Tts-gan: A transformer-based time-series generative adversarial network," in *Artificial Intelligence in Medicine*, M. Michalowski, S. S. R. Abidi, and S. Abidi, Eds., Cham: Springer International Publishing, 2022, pp. 133–143, ISBN: 978-3-031-09342-5.

[15] X. Zhang, J. Shi, M. Yang, *et al.*, "Real-time pipeline leak detection and localization using an attention-based lstm approach," *Process Safety and Environmental Protection*, vol. 174, pp. 460–472, 2023, ISSN: 0957-5820. DOI: https://doi.org/10.1016/j.psep.2023.04.020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957582023003087.

[16] T. Sun, X. Wang, J. Wang, *et al.*, "Magnetic anomaly detection of adjacent parallel pipelines using deep learning neural networks," *Computers Geosciences*, vol. 159, p. 104 987, 2022, ISSN: 0098-3004. DOI: https://doi.org/10.1016/j.cageo.2021.104987. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0098300421002715.

[17] Z. Liao, H. Yan, Z. Tang, X. Chu, and T. Tao, "Deep learning identifies leak in water pipeline system using transient frequency response," *Process Safety and Environmental Protection*, vol. 155, pp. 355–365, 2021, ISSN: 0957-5820. DOI: https://doi.org/10.1016/j.psep.2021.09.033. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957582021005061.

[18] G. An, Z. Huang, and Y. Li, "Constant false alarm rate detection of pipeline leakage based on acoustic sensors," *Scientific Reports*, vol. 13, no. 1, p. 14 149, 2023, ISSN: 2045-2322. DOI: 10.1038/s41598-023-41177-3. [Online]. Available: https://doi.org/10.1038/s41598-023-41177-3.

[19] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 421–425. DOI: 10.1109/ICASSP.2017.7952190.

[20] W. Zha, Y. Liu, Y. Wan, *et al.*, "Forecasting monthly gas field production based on the cnn-lstm model," *Energy*, vol. 260, p. 124 889, 2022, ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2022.124889. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544222017923.

[21] P. Nandi and M. Rao, "A novel cnn-lstm model based non-invasive cuff-less blood pressure estimation system," in *Annu Int Conf IEEE Eng Med Biol Soc*, Jul. 2022, pp. 832–836. DOI: 10.1109/EMBC48229.2022.9871777.

[22] H. Abbasimehr and R. Paki, "Improving time series forecasting using lstm and attention models," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 1, pp. 673–691, 2022.

[23] P. Li, X. Jiang, G. Jin, Y. Yu, and Z. Xie, "Alstm: An attention-based lstm model for multi-scenario bandwidth prediction," in *2021 IEEE 27th International Conference on Parallel and Distributed Systems (ICPADS)*, 2021, pp. 98–105. DOI: 10.1109/ICPADS53394.2021.00018.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[25] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. DOI: 10.1038/323533a0.

[26] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[27] High Fidelity Engineering, *High Fidelity Engineering*, https://hifieng.com/, Accessed on January 4, 2024, 2023.