Hindawi
Security and Communication Networks
Volume 2023, Article ID 8903980, 10 pages
https://doi.org/10.1155/2023/8903980

Research Article

An Anomaly Detection Approach Based on Integrated LSTM for IoT Big Data

Chao Li,[1] **Yuhan Fu**,[1,2] **Rui Zhang**,[3] **Hai Liang**,[2] **Chonghua Wang**,[4] **and Junjian Li**[1]

[1] *Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510700, China*
[2] *Guangxi Key Laboratory of Cryptography and Information Security, Guilin 541004, China*
[3] *COMAC Shanghai Aircraft Manufacturing Co., Ltd, Beijing 100005, China*
[4] *China Industrial Control Systems Cyber Emergency Response Team, Beijing 100040, China*

Correspondence should be addressed to Chonghua Wang; chonghuaw@live.com

Received 17 October 2022; Revised 8 December 2022; Accepted 25 April 2023; Published 30 May 2023

Academic Editor: Yulei Wu

Due to the expanding scope of Industry 4.0, the Internet of Things has become an important element of the information age. Cyber security relies heavily on intrusion detection systems for Internet of Things (IoT) devices. In the face of complex network data and diverse intrusion methods, today's network security environment requires more suitable machine learning methods to meet its security needs, and the current machine learning methods are hardly competent. In part because of network attacks by intruders using cutting-edge techniques and the constrained environment of IoT devices themselves, the most widely used algorithms in recent years include CNN and LSTM, with the former being particularly good at extracting features from the original data space and the latter concentrating more on temporal features of the data. We aim to address the issue of merging spatial and temporal variables in intrusion detection models by introducing a fusion model CNN and C-LSTM in this paper. Fusion features enhanced parallelism in the training process and better results without a very deep network, giving the model a shorter training time, fast convergence, and computational speed for emerging resource-limited network entities. This model is more suitable for anomaly detection tasks in the resource-constrained and time-sensitive big data environment of the Internet of Things. KDDCup-99, a publicly available IBD dataset, was applied in our experiments to demonstrate the model's validity. In comparison to existing deep learning implementations, our proposed multiclass classification model delivers higher accuracy, precision, and recall.

1. Introduction

Internet of Things (IoT) is a new network system consisting of a cloud data center and subnodes under it that integrates computing, controlling, and communication technologies. In the era of industry 4.0, wireless network technology and diverse smart devices are increasingly applied to the Industrial Internet of Things (IIoT), and more and more industrial applications are interactively connected through the intelligence and real time of signal processing. Through a large number of distributed IoT devices, ubiquitous sensors are deployed throughout real scenarios. They detect environmental data through various types of sensors and transmit them to processing centers through various types of IoT transmission protocols. The processing center uses cloud computing and big data technologies to extract valuable information from this data and upgrade services. IoT has been frequently employed in various fields such as healthcare, smart home, and intelligent transportation. By 2024, IoT is anticipated to reach 83 billion devices [1]. The diverse category of IoT devices will set off the IoT architecture for innovation.

In addition, cybercrime is growing dramatically in size, complexity, and cost [1] due to the increasing spread of IoT devices with distributed and large numbers in individual homes, national grids, smart cars, and industrial assembly lines, and the complexity of IoT defending systems [2]. Table 1 lists several typical cyber-attacks. The rise of various old and new types of cyber-attacks signifies that the use of resume firewalls and signature certificate-based defending is

Table 1: Several common types of network attacks.

Attacks	Quantity
DoS and DDoS	DoS attack is designed to overload system resources to the point where they can no longer respond to legitimate service requests. And DDoS is initiated by controlling a large number of hosts infected with malware
MITM	As an "indirect" intrusion attack, a man-in-the-middle (MITM) type of network attack allows an attacker to eavesdrop and steal communications from two computers without directly affecting the network
DNS spoofing	Spoofing through the domain name system (DNS) is also a form of man-in-the-middle attack, where a hacker can change the DNS records returned to the querier to a response record of the attacker's choosing
URL resolution	Through URL interpretation, an attacker can change and forge certain URL addresses and access to personal or company private data
Zero-day attacks	Zero-day attacks are computer vulnerabilities that have not been discovered by security vendors but may be in the hands of hacker groups, and once they are discovered, 0 day vulnerability attacks can spread rapidly

in great demand, and instead, a proactive approach must be taken to discover threats. Intrusion detection systems have become a crucial tool for identifying and defending against network attacks in the form of malicious network traffic as security threats continue to spread across the Internet. By extracting features for analysis of network traffic and alerting once unsafe traffic is detected, intrusion detection systems enable network monitoring [3].

Anomaly detection is recognized as one of the key tools for dynamic network security threat detection [4]. There are numerous methods available for network anomaly detection. To improve the performance of anomaly detection systems when processed by intrusion detection systems, artificial intelligence techniques are applied to various types of active-defending systems. However, reliable anomaly detection of massive and complex multidimensional data in industrial IoT is still a tricky task. In recent years, deep learning has excelled in various classification tasks, but a large amount of classification variables in the network stream complicates the anomaly detection process using gradient descent methods. Even though there are numerous methods for anomaly detection, most people do not try to use CNN (convolutional neural network) for anomaly detection, compared to machine learning. As research has intensified in recent years, deep learning has increasingly emerged in the field of complex high-level data processing, such as image and signal processing. Deep learning interprets the internal rules and data expressions of data such as word, image, and sound by extracting the internal features of sample data during the learning process. The ultimate goal is that deep learning models have the ability to analyze and learn from input data and eventually recognize data such as characters, images, and sounds. Among them, two models are widely used: recurrent neural networks (RNN) that mainly extract time-step features for problems of NLP and voice recognition, and CNN with powerful spatial feature extraction for image classification and regression. Zeiler [5] visually understands the functions of the intermediate feature layer and the operation of the classifier through the large convolution neural network model, indicating that CNN is very sensitive to the local structure of data. CNNs reflect the spatial properties of data by extracting spatial cues such as color, level, and edges in images using convolutional perceptual fields and shared weight coefficients while RNN uses gate units to efficiently simulate serialized data to reflect the temporal properties of the data. The LSTM method is mentioned in all model surveys for univariate and multivariate time series data mentioned by Lindemann et al. [6]. To achieve high performance, Kim and Cho [7] constructed a C-LSTM network. They first used preprocessing to initially construct temporal correlations of the dataset, then used CNN to extract these features, and finally used LSTM to extract spatial and temporal features. To ensure that the features extracted by CNN are potentially correlated and more effective than the temporal features extracted by LSTM, Preciado-Grijalva and Iza-Teran [8] used two sliding windows to generate time-dependent subsequences based on C-LSTM. Meanwhile, Yin employed a modified LSTM-based self-encoder to extract more anomalous features from the input sequence. Although CNN and LSTM are both part of their network, the input that LSTM accepts only comes from CNN extraction, and the spatialized extraction of CNN disrupts the temporal aspects of the original data at the potential level, which impacts LSTM's learning effect to some extent.

We propose an improved network structure based on the study of the interaction between CNN and LSTM direct serial methods for extracting data features. The network consists of a CNN and a C-LSTM using temporal convolution, both of which receive input from the original dataset, and the CNN and C-LSTM will focus on extracting spatio-temporal features in the intrusion data, respectively, with the modified C-LSTM using one-dimensional temporal convolution and the LSTM focusing more on purely temporal features of the intrusion data while ignoring some spatial features; the CNN will learn more to reconstruct the spatial features of the intrusion data and do parallel and fusion between the two instead of serial, which can improve the performance of the detection model. In the absence of deeper network depth, anomaly detection achieves higher scores in various metrics. The significant contributions of this paper are as follows:

(1) A network model based on our C-LSTM and CNN fusion is proposed to detect intrusion data using shallow, small-scale deep learning models. The model utilizes the ideas of C-LSTM and exploits its advantages. To retrieve temporal aspects of the intrusion data on a partially parallel model, a one-dimensional convolution is employed in place of the two-dimensional convolution in the original C-LSTM.

(2) Fusing our C-LSTM and CNN to obtain a balance on temporal and spatial features. Compared to the original C-LSTM which presents the final prediction results and scores by a single model, our fusion model is trained and predicted independently by two models. The learning method of fusion learning determines that it does not extract features by CNN and then learn features by LSTM but fuses the features learned by C-LSTM and CNN separately, which extends the dimensionality of the features. In this way, even if the features extracted by the CNN are insufficient, the C-LSTM can be supplemented and extended. In terms of the evaluation of several classification metrics, this model outperforms the C-LSTM model.

(3) In light of the model fusion learning method, its two-part model does not require a deep model depth to learn every feature of the entire dataset and is faster in its training and convergence than other methods. The two parts of our network, C-LSTM and CNN, only need to learn the sensitive part of the data features and combine them for prediction, rather than learning all the features separately. Such a mechanism facilitates its performance on resource-constrained devices.

2. Related Work

This section summarizes machine learning methods for anomaly detection based on a review of researchers' research on anomaly detection or intrusion detection in recent years. As shown in Table 2, in the field of anomaly detection, many researchers classify a segment of the data as normal or anomalous by extracting contextually relevant information from the data. However, in general, this discriminative approach also requires modeling of the data system, so detection methods are divided into three categories: modeling based on data statistics, modeling based on temporal features, and modeling based on spatial features. Either the error in the context of the data predicted by the time series is used as the core of detection [9], or the original data is reconstructed without a priori knowledge, and normal and abnormal values are defined using thresholds [10]. The core of all these detection methods is to extract the necessary features from the original data space that affect its determination as normal or abnormal and thus perform a credible classification.

2.1. Issues on Intrusion Detection System. Researchers have already done in-depth research in the area of intrusion detection for cybersecurity of IoT, cloud data centers, and blockchain systems. They have investigated the general anomaly detection problem by discussing anomaly detection in time-series data in short chapters.

Table 2: Several abnormal traffic detection methods.

Authors	Model	Dataset	Year	Score
G. Bae	CNN	KDD99	2019	Acc = 97.34
A. Diro	LSTM	AWID	2018	Acc = 98.22
Q. Tian	Svm	UNSW-NB15	2019	Acc = 97.00
Y. N. Kunang	DNN	NSL-KDD	2021	Acc = 83.33
In-young	C-LSTM	Webscope	2018	Acc = 99.62
Chunyong Yin	C-LSTM-AE	Webscope	2021	Acc = 98.6

Hawkins [11] and Abraham and Chuang [12], as early developers in this field, have conducted in-depth research on the network security of wireless network, Internet of Things, blockchain system, and other network systems, especially the network security intrusion problem. However, many researches on anomaly detection have found this kind of problem and usually discuss the intrusion problem highlighted by abnormal data in several sections. Markou and Singh [13, 14] have published research showing that among the intrusion detection methods up to 2003, the intrusion detection system based on feature extraction has been widely used these days. Stephen and Arockiam [15] designed a protocol suitable for resource-constrained nodes but with lossy routing and explored an integrated approach to detect Sybil attacks on the IoT.

2.2. Unsupervised Method in Anomaly Detection. Münz et al. used the unsupervised learning method of K-means [16]. They discussed and derived the prime number of clusters by statistical methods. By calculating the prime number and the spatial distance of each flow data, the distance data are used as the standard for distinguishing abnormal and normal data. Zhang and Zulkernine adopted the random forest method based on unsupervised learning [17], calculated the closeness in each case, and designed a mathematical standard based on statistics to distinguish normal and abnormal data.

2.3. Machine Learning in Anomaly Detection. Kaur et al. [18] used CNN models to detect attacks in data streams. They trained and validated their model with the cicids2017 and cicids22018 datasets. Although their approach covers a wide range of intrusion data types, their performance metrics fall short of practicality. To detect intrusions in a massive data environment, Hassan et al. [19] designed an integrated deep learning model using CNN and *wdlstm* (long-term memory with decreasing weights). CNN was used to find the best features, and *wdlstm* method was used to prevent overfitting in neural networks [20]. The Bayesian neural network is studied in, and the LSTM based self-encoder is used to replace some previous data extraction and analysis structures, and then, the MLP is used to perform the final prediction step. Chen and Lin constructed time-step features of the original data using a sliding window preprocessing algorithm. Then LSTM models were used to extract information on the preprocessed high-dimensional data [21]. The LSTM model studied by Malhotra et al. [9] was based on sensor data and normal signals. They used the trained LSTM model to predict the succeeding signal as a criterion for judging, and then, the actual input and the criterion were calculated to derive the error distribution for anomaly detection.

These methods use models that model serialized data to learn temporal features and thus have the ability to predict predictive classification. Just like RNN, after training RNN, RNN will have the ability to predict future data. In this way, the output of the model can be used as a criterion to compare with the actual data. The result of the comparison will be determined by a set threshold value to determine whether it is normal or abnormal. This approach is based on periodic traffic data and performs better on its dataset. However, if periodicity is not predominantly represented on the data set, the predicted results will not be accurate for the actual data.

As mentioned previously, many attempts have been made to detect intrusion data using various methods. However, few attempts have been made to focus on spatio-temporal features of the data to conduct research. Most studies, in order to improve the performance of a single dataset, usually model only the key features in the datasets that have a high degree of impact, while few studies have talked about taking both temporal and spatial features into account. In contrast, spatio-temporal information is crucial for data analysis and reconstruction because it integrates the spatio-temporal features of the original data. In order to utilize both temporal and spatial information in complex traffic data for anomaly identification, a suitable learning method is therefore required.

3. Modeling of C^2-LSTM

Using models that make compromise judgments on temporal and spatial features, C^2-LSTM is a modified C^2-LSTM model designed to evade attacks to achieve deception of resource-constrained models by intrusion data. In the C^2-LSTM, the CNN and the improved C^2-LSTM learn the spatial and temporal dimensions of the intrusion traffic, respectively.

3.1. Problem Definition. Set $D = \{X^1, X^2, \ldots, X^k\}$ is a input set which represents k kinds of labeled network traffic data in anomaly detection. k includes q kinds of anomaly samples and p kinds of normal samples.To be specific, p is much bigger than q, and $p + q = k$. $Y = \{y^1, y^2, \ldots, y^k\}$ is a set of label results for input D. The work in [22–24] tries to find outliers in the data and edit to identify them, while the work in [25, 26] is for the classification and labeling of the target sequence, whether it is abnormal or normal. The former is a regression method that regresses the input data into an exact value. The latter is a clustering or classification problem to classify it into one of the predefined categories. The issue studied in our paper is a classification problem, i.e., classifying the input data into a corresponding type $y^\imath$.

3.2. Our C-LSTM. CNN and LSTM are the two components of C-LSTM. Similar to C-LSTM, he uses self-encoders based on LSTM and CNN to extract fused spatio-temporal features from the data. Figure 1 shows its model structure.

The C-LSTM uses preprocessed data as input. The convolutional layer uses convolutional kernels for learning and feature extraction, and the parameters of each layer are optimized by a back-propagation algorithm. Convolutional operations can extract various features from the data space level. The first few layers of convolution may only extract some low-level features. For images, these are picture corners, single lines, edges of objects, etc. that are not sensitive to the impact of the results, while higher level features that affect the model performance will be extracted north in the deeper layers of the network. The pooling layer reduces the computational effort by partitioning and sampling the data, down sampling a large matrix into a smaller one, and can prevent overfitting at the same time. The feature data are transported to the LSTM. First, the CNN is composed of a convolutional layer and a pooling layer for automatically extracting a sequence of high-level spatial features of the network traffic. We use a one-dimensional convolution operation to extract the temporal features of the input data directly by temporal convolution instead of the normal two-dimensional convolution of C-LSTM. After the convolution, an activation function is used to perform the transformation of the non-nonlinear function. As a result, the model is able to capture features of more dimensions.

Suppose it is an input vector of intrusion data and n is the dimensionality of its features. Equation (1) yields the output value from the i-th convolutional layer.

$$y_i = \sigma\left(b_i + W_i \bullet x\right), \tag{1}$$

where b is the bias of the feature mapping, W is the weight of the kernel, and σ is an activation function.

We use circular units running from left to right to enable the LSTM layer to understand the temporal properties of the traffic data extracted from the upper CNN layer. This makes the model in this layer to have a stronger understanding of the feature transformation relationships on the time scale. His input is the output of the pooling layer of the upper layer, which is gated to control the discarding or adding of information for forgetting or remembering. Gating is an information selective pass-through structure based on a multiplicative mechanism, consisting of a sigmoid function and a dot product operation that updates the cell state of each gate according to its activation. Sigmoid functions have output values in the interval [0, 1], with 0 representing complete discard and 1 representing complete pass-through. Cell management through these gates handles the upper layer of input to input, output and forgetting gate operations. The hidden value of the LSTM cell, h_t, is updated once per step t.

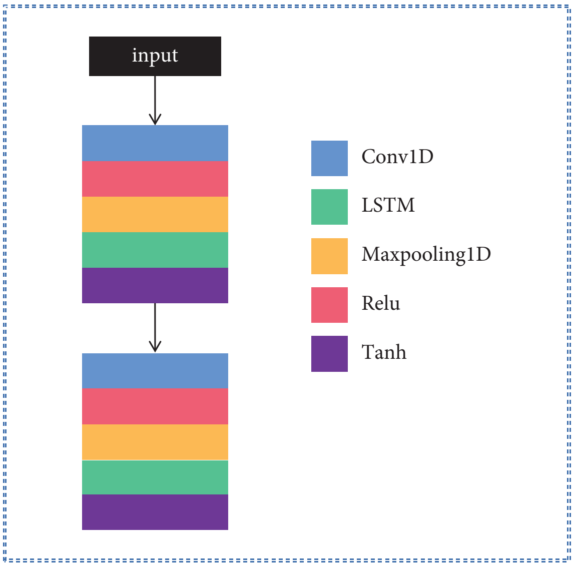


Figure 1: Model structure of our C-LSTM which consists of one-dimensional convolution, LSTM, and one-dimensional maxpooling.

$$f_t = \sigma_g\left(W_f x_t + U_f c_{t-1} + b_f\right), \tag{2}$$

$$i_t = \sigma_g\left(W_i x_t + U_i c_{t-1} + b_i\right), \tag{3}$$

$$o_t = \sigma_g\left(W_o x_t + U_o c_{t-1} + b_o\right), \tag{4}$$

$$c_t = f_t \bullet c_{t-1} + i_t \bullet \sigma_c\left(W_c x_t + b_c\right), \tag{5}$$

$$h_t = o_t \bullet \sigma_h\left(c_t\right). \tag{6}$$

Equations (2)–(4) use the symbol I: update the gate, take sigmoid for the splicing result to indicate whether the previous result needs to be updated, F: forget the gate, take sigmoid for the splicing result to indicate whether the previous result is discarded and O: synchronize the gate, take sigmoid to indicate whether synchronization is required. C and h, which stand for cell states and hidden values, respectively, are used in equations (5) and (6). The forgetting gate, the input gate, and the output gate work together to calculate these two values. These gates have only few linear interactions with the other parts. σ is an activation function. The network using LSTM units provides excellent learning capabilities through modeling signal time feature, which yields the most advanced results in anomaly detection.

3.3. Fusion Model C^2-LSTM.

C^2-LSTM uses a CNN and C-LSTM for fusion. For the purpose of extracting spatio-temporal features from the intrusion data center, CNN and C-LSTM are used as two independent strong learners. Anomaly detection is performed by splicing the two features through fusion. The two learners work parallelly to generate and evaluate the final result; or in other words, there is no reliance between them. Figure 2 illustrates its model structure.

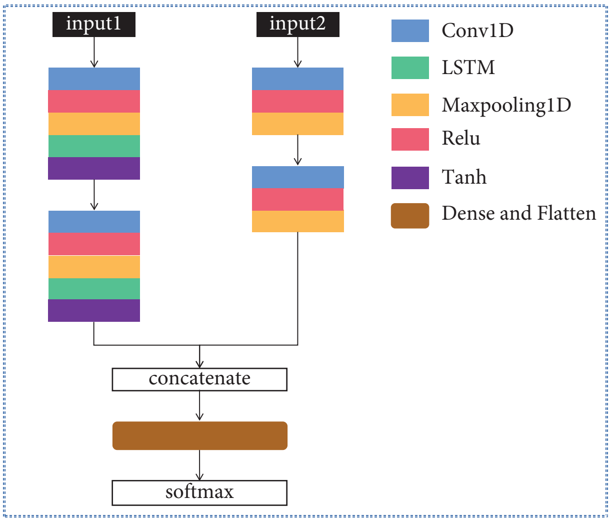


Figure 2: Model structure of our C^2-LSTM which consists of a CNN and our C-LSTM.

The features retrieved by the CNN and the C-LSTM are fused by fusion, and the features are stitched together after both the CNN and LSTM have derived their own values.

$$S_1 = \sigma\left(b_i + W_i \bullet x\right), \tag{7}$$

$$S_2 = o_t \bullet \sigma_h\left(c_t\right), \tag{8}$$

$$S = \left[\left[S_1\right], \left[S_2\right]\right], \tag{9}$$

where S_1 is the output of the C-LSTM, and S_2 is the output of the LSTM. S is the stitching of the two feature matrices in the last dimension. Here, we use the concatenate method to blend features and models. At the end of the model, a concatenate layer is built to combine the features extracted from the previous model. We spliced the tensors in the last dimension and ensured the alignment of the two parts of features in the last dimension. This makes the fused features rely on more than just the results of the previous operation step, combining features of different properties.

In traffic detection, the fully connected layer is responsible for reducing the sensitivity of the parameters in the learning process. And *Softmax* is used to output the final classification score. They are the layers used for the output of the C^2-LSTM model. In the upper part, the output of the fusion matrix is stretched, and this vector will be fed to the fully connected layer. This layer uses equation (10). D denotes the output of the fully connected layer, and σ is the activation function.

$$D_i = \sum \sigma\left(b_i + W_i \bullet h\right). \tag{10}$$

The output of the fully connected layer is multiclassified by *softmax* and *softmax* layer classifies the raw data into normal and abnormal classes.

3.4. Schema and Super Parameters.

The input of C^2-LSTM is $41 * 1$ traffic input, and the parameters of various types of structures can be adjusted under the design conditions of the model. The input of the C2-LSTM is $41 * 1$ traffic data. Under

```
Input: $D_1 = \{X_1^1, X_1^2, \ldots, X_1^k\}, D_2 = \{X_2^1, X_2^2, \ldots, X_2^k\}$ are the input sets, and label $Y = \{y^1, y^2, \ldots, y^k\}$ is the corresponding
Output:A trained anomaly detection model $M$
Initialize the model $M$
Initialize the iteration count $T$, batch size $N$, threshold $\delta$
for $q = 1$ t o $T$ do
    for $m = 1$ to 2 do
        for each batch $\{X_m^i\}_{i=1}^N$ do
            Transfer $X_1^i$ into $S_1$ via CNN by equation (7)
            Transfer $X_2^i$ into $S_2$ via CNN by equation (8)
            Splice $S_1$ and $S_2$ into $S$
            Predict $y^{(i)}$ based on $Z$ via the estimation network
            Update $M$ to minimize loss
    end for
    if loss < $\delta$: break
end for
return $M$
```

Algorithm 1: Anomaly Detection Algorithm Based on C^2-LSTM.

the design conditions of the model, the parameters of various types of structures can be adjusted, such as the depth of CNN and LSTM, the design of convolution, and the gating strategy of LSTM. These settings will determine the final performance of the whole model, such as accuracy or learning speed. In contrast, the C^2-LSTM fusion method determines that it does not require a high number of layers. Before entering the LSTM, he becomes $19 * 32$ spatially feature-rich data by convolution layer and pooling layer. We use ReLU as the activation function of the model. After recent years of research, ReLU learns faster than Tanh, and he ensures that the product of the parts is all always 1, and there is no problem of gradient disappearance of the sigmoid.

3.5. C2-LSTM Anomaly Detection Algorithm. The workflow of C^2-LSTM anomaly detection is as follows: first, preprocess and normalize sample raw data to handle dirty data and construct input sets X1 and X2. Then, the input datasets are learned with CNN and C-LSTM, respectively, and the outputs of both are two feature matrices S_1 and S_2. By dimensional stitching of feature fusion, S_1 and S_2 are reconstructed into S. The reconstructed features S will be used in the final prediction to get the final classification of that traffic. The specific algorithm is shown in Algorithm 1.

4. Experiment and Analysis

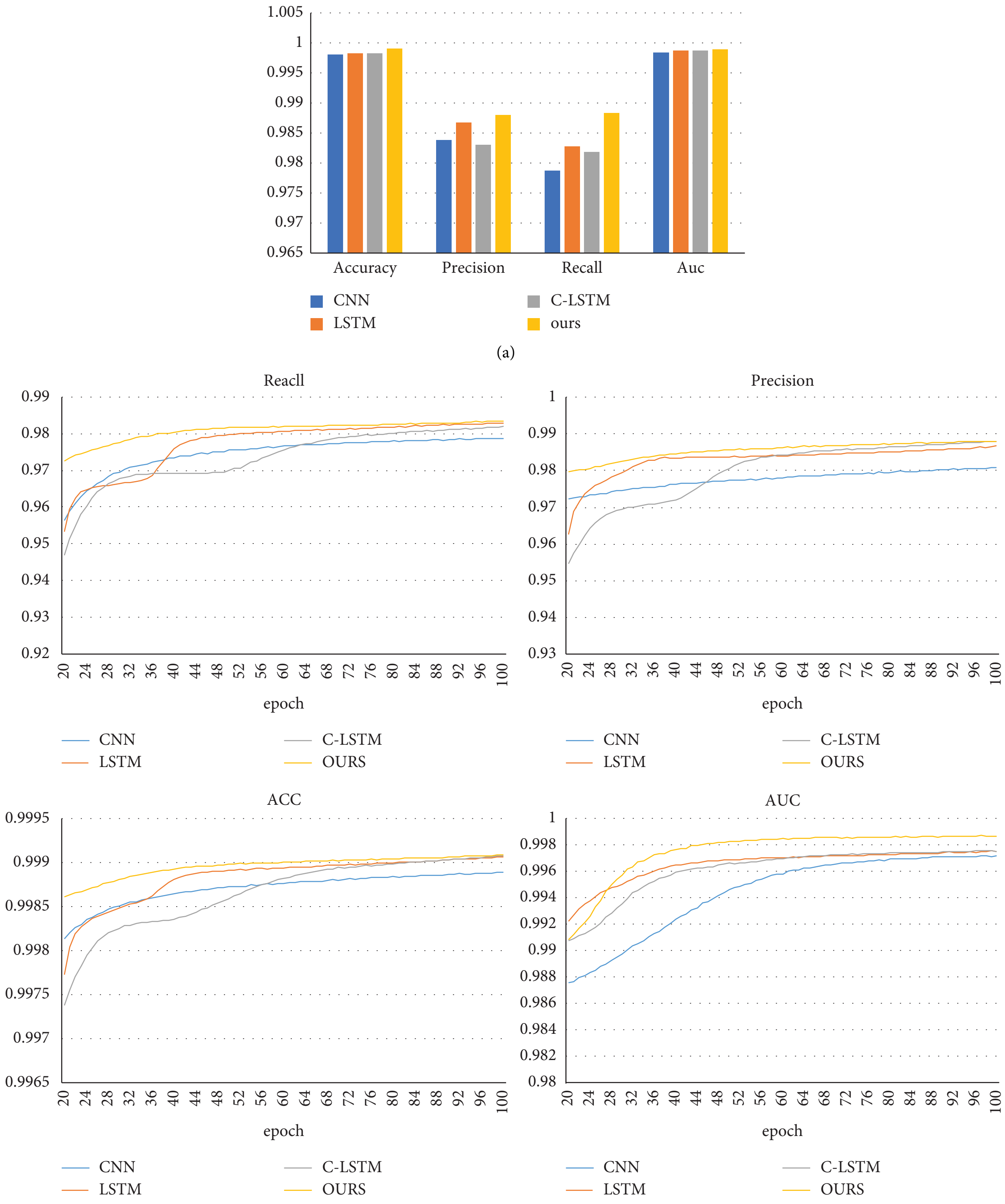
We use Python 3.7 as the programming language and the CUDA version of Tensorflow 1.14.0 as the neural network framework. The lab deploys comparison experiments in the same environment and trains models on 4 GeForce RTX 2080 Ti with 12G video memory. We use small batch training in the experiments with a batch size of 512. The details of the experiments are as follows.

4.1. Data Sets and Preprocessing. KDD99 is a dataset for monitoring abnormal connections from normal connections, from the DARPA Intrusion Detection Evaluation Project in 1998. The KDD99 dataset is a feature extract version of the DARPA dataset (DARPA is the original dataset), and the training data for the experiment were 7 weeks of network traffic. This dataset was utilized in the KDDCUP competition in 1999 and later became known as the KDD99 dataset. Although the dataset is too old and may have obsolescence issues, KDD99 was very popular among researchers and sets the stage for deep learning and intelligent computing to make a big splash in intrusion research.

Each entry in the dataset is labeled, specifically into 2 types of anomalous attacks and 1 type of normal. We train a random sample at a time to learn the characteristics of each anomaly type in order to make predictions for each input data.

In the experimental study, the network intrusion detection packet kdd_cup_data_10percent from KDDCup99 is marked as the training set and corrected as the test set. The kddcup_data_10percent packet is a 10% sample of the kddcup_data packet. Since the data processed for the experiment is network traffic, inputting a segment of network traffic predicts the category to which it belongs (39 attacks + normal). For such a classification problem, we conducted similar experiments in different models and evaluated these models by accuracy, precision, recall, and f1 score.

To explain our evaluation metrics, the following explanation is given. Suppose a correct sample is incorrectly considered as wrong in a dichotomous classification problem, and this wrong data is labeled as false positive (FP). A false negative (FN) indicates that an abnormal instance is labeled as normal. Similarly, true positives (TP) and true negatives (TN) indicate abnormalities and correctly identify normal instances. The area enclosed by axes under the ROC curve is defined as the AUC (area under the curve), which has values between 0.5 and 1 in a $1 * 1$ coordinate system. The closer the AUC is to 1.0, the better the prediction equals to 0.5, the lowest truthfulness and no application value. Different metrics can be evaluated in this way:



(a)

(b)

Figure 3: (a) Comparison of the metrics on the test data set and (b) comparison of the metrics on 100 epochs during training.

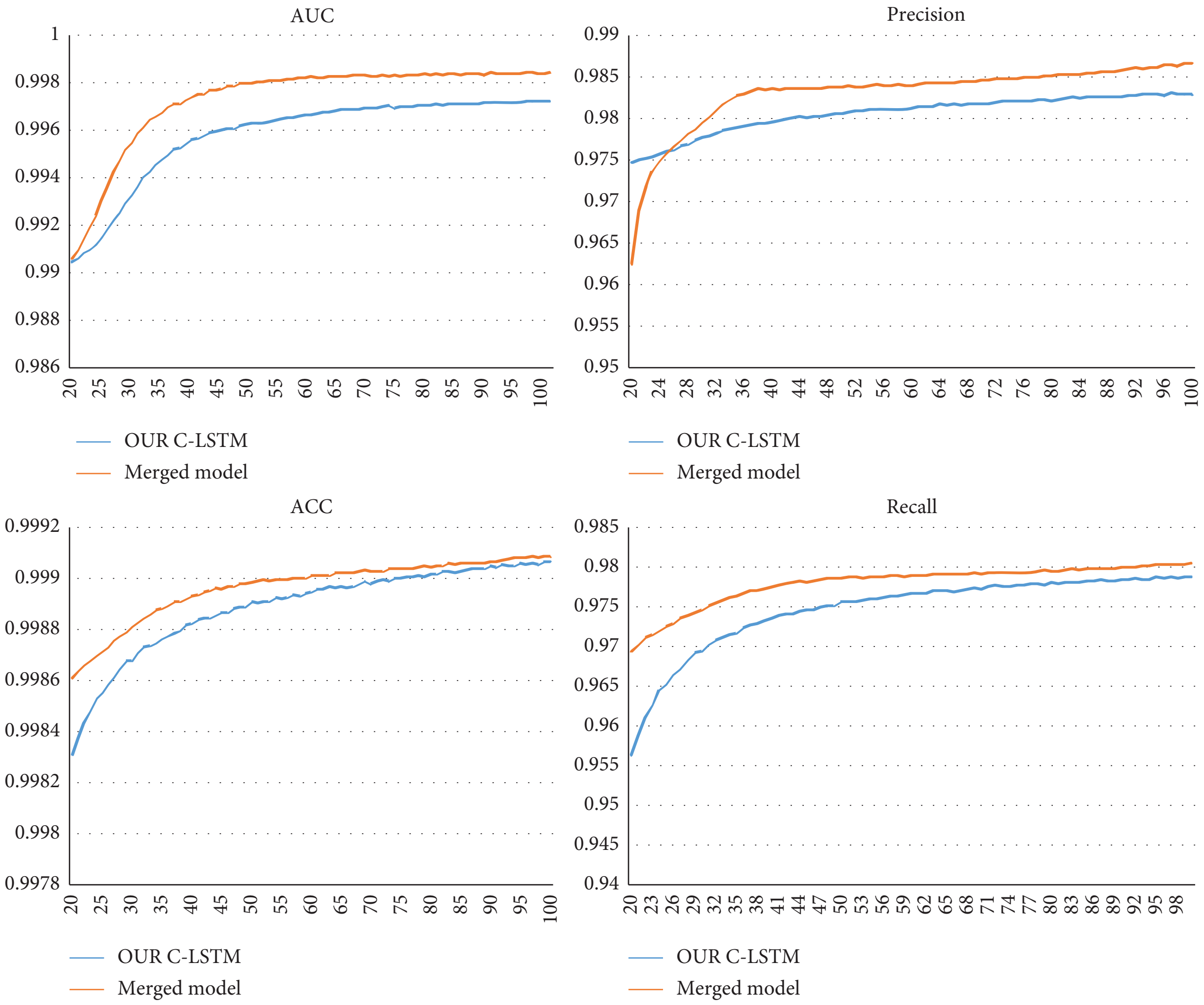


Figure 4: Train these two models with 100 epochs of training data for comparison.

$$
\begin{aligned}
\text{accuracy} &= \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \\
\text{precision} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \\
\text{recall} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \\
\mathrm{AUC} &= \frac{1}{2}\sum_{i=1}^{m-1}\left(x_{i+1} - x_i\right)\bullet\left(y_{i+1} + y_i\right).
\end{aligned}
\tag{11}
$$

Data preprocessing using python consists of numerical replacement text, numerical normalization, and tag unique hot coding. Numerical replacement text mainly converts the values of the 41 feature values of each connection that are strings into numerical form. The most-valued normalization is used in numerical normalization. The preprocessing ends into 4 files (train_x, train_y, test_x, test_y).

4.2. Analysis of Indicators. To build our model and conduct experiments, we utilized two models; the CNN can extract spatial features up to dimensionality, while the C-LSTM is more sensitive to changes on time steps. We designed experiments to evaluate the effect of CNN and C-LSTM fusion. We set the number of convolution cores of the convolution layer to 32 and the step size to 3. The window size of the pooled layer is 2. The number of LSTM cells is 64. In this experiment, we evaluated four models including CNN, LSTM, C-LSTM, and C^2-LSTM. After each model was trained on the training dataset for 100 calendar hours at a learning rate of 1e-2, the performance (precision, accuracy, recall, and Auc score) on the test dataset was collected as shown in Figure 3. The fusion model with C^2-LSTM has the highest scores in terms of training accuracy, precision, recall, and Auc. From the test results, we can conclude that our C-LSTM outperforms the single CNN and LSTM. Proving that it can extract more key features in time series, our C-LSTM also demonstrates better temporal feature extraction and is higher than the original LSTM in terms of AUC, accuracy, and precision metrics compared to the single LSTM. At the same time, it is slightly lower than the LSTM in terms of recall metrics, which we believe is due to the one-dimensional CNN in front of the model that makes it focus more on single temporal features and ignore spatial features in some dimensions.

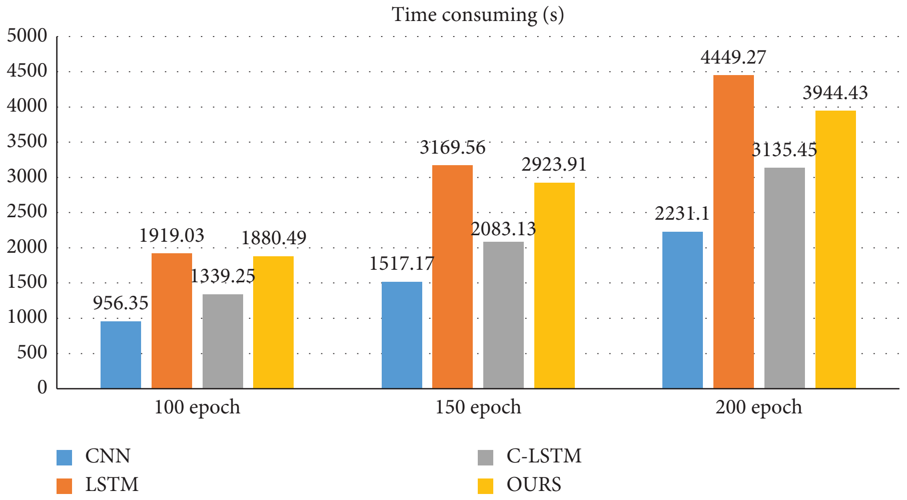


Figure 5: Convergence speed of training.

4.3. Comparison with Our C-LSTM. In the paper [7], the authors use sliding windows to construct preprocessed temporal correlation data and use LSTM to extract temporal features. In order to achieve better results in temporal feature extraction with the fused model, our improved model replaces the C-LSTM's two-dimensional convolution and sliding window operation for extracting temporal features with a one-dimensional temporal convolution and fuses it with the CNN. To demonstrate the effectiveness of the improvement and fusion, we take out the improved model separately and compare it with the C^2-LSTM. We evaluate the classification results of the C-LSTM model with one-dimensional convolution and the C^2-LSTM. As shown in Figure 4, the two models are trained with 100 epochs of training data with a learning rate of 1e-3. The lowest loss values and best performance across all metrics are obtained by the fused C^2-LSTM. The fused C^2-LSTM based on fusion can extract superior features for further classification, according to the experiments.

4.4. Time Performance Requirements. Driven by the new digital revolution represented by IoT technologies, these emerging resource-constrained network entities usually have limited computing power or more sensitive latency characteristics. Therefore, the training and detection time of the network and its own lightweight are also discussed in our consideration and comparison experiments. We compared the training time and prediction time for CNN, LSTM, our modified C-LSTM, and our fused C^2-LSTM. The experimental conditions were a GeForce RTX 2080 Ti with 12G video memory. The training time is the total training time when training 100 epochs with a batch size of 512, and the prediction time is the overall prediction time when predicting 494021 data using the trained model. Figure 5 depicts the performance at training time, with the fused C^2-LSTM converging fastest at a lower loss. Our C^2-LSTM offers faster prediction and training speed compared to simple CNN, as seen in Table 3. This is due to the fact that our fusion model possesses superior capability in prediction without requiring deep network layers.

Table 3: Time consumption for training and prediction.

	CNN (*s*)	LSTM (*s*)	Our C-LSTM (s)	Ours (s)
Training time	956.35	1919.03	1339.25	1880.49
Testing time	3.06	3.09	2.81	2.62

5. Conclusions

We have proposed a new architecture combining CNN and enhanced C-LSTM to better adapt to the emerging IoT anomaly intrusion detection with massive data and high latency sensitivity. In addition, we demonstrate that the architecture that extracts spatial and temporal features separately in parallel from CNN and C-LSTM before fusing them can better learn both spatial and temporal correlations of data simultaneously to better cope with complex IoT environments. Based on this, we have evaluated different anomaly detection methods and used C^2-LSTM to extract superior features for classification in fully connected networks. According to the results of the experiments, the model has performed at the highest level in terms of accuracy, precision, completeness and AUC score. Furthermore, its model structure determines that it can boost detection performance without a deep network and can also evaluate temporal performance at a higher level. It is challenging to sustain its existing edge over shallow networks in the face of ultrahigh latitude data, though, as the complexity of the data keeps growing. When faced with such data, we have intended to use PCA to downscale and process the data, but using data preprocessing methods will inevitably introduce some latency, which is not permitted in industrial IoT devices listed with high latency sensitivity, and we will continue to work in this direction in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Key R&D Program of China (no. 2022YFB3104100), the Major Research Plan of the National Natural Science Foundation of China (no. 92167203), the Guangzhou Science and Technology Plan Project (no. 202201020216), and the Guangxi Key Laboratory of Cryptography and Information Security (no. GXIS202119).

References

[1] R. Vinayakumar, M. Alazab, S. Srinivasan, Q. V. Pham, S. K. Padannayil, and K. Simran, "A visualized botnet detection system based deep learning for the internet of things networks of smart cities," *IEEE Transactions on Industry Applications*, vol. 56, pp. 4436–4456, 2020.

[2] S. Smith, "IoT connections to reach 83 billion by 2024, driven by maturing industrial use cases," 2020, https://www.juniperresearch.com/press/iot-connections-to-reach-83-bn-by-2024#:%7E:text=Industrial%20Use%20Cases-,IoT%20Connections%20to%20Reach%2083%20Billion%20by%202024%2C%20Driven%20by,35%20billion%20connections%20in%202020.

[3] R. Vinayakumar, M. Alazab, K. P. Soman, P. Poornachandran, A. Al-Nemrat, and S. Venkatraman, "Deep learning approach for intelligent intrusion detection system," *IEEE Access*, vol. 7, pp. 41525–41550, 2019.

[4] E. Schubert, A. Zimek, and H. P. Kriegel, "Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection," *Data Mining and Knowledge Discovery*, vol. 28, pp. 190–237, 2014.

[5] M. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proceedings of the Computer Vision-ECCV 2014 13th European Conference*, Zurich, Switzerland, September 2014.

[6] B. Lindemann, B. Maschler, N. Sahlab, and M. Weyrich, "A survey on anomaly detection for technical systems using LSTM networks," *Computers in Industry*, vol. 131, Article ID 103498, 2021.

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

[8] A. Preciado-Grijalva and V. R. Iza-Teran, "Anomaly detection of wind turbine time series using variational recurrent autoencoders," 2021, https://arxiv.org/abs/2112.02468.

[9] P. Malhotra, L. Vig, and G. Shroff, "Long short-term memory networks for anomaly detection in time series," in *Proceedings of the 31th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, vol. 89, Bruges, Belgium, December 2015.

[10] P. Malhotra, A. Ramakrishnan, and G. Anand, "LSTM-based encoder-decoder for multi-sensor anomaly detection," 2016, https://arxiv.org/abs/1607.00148.

[11] D. M. Hawkins, *Identification of Outliers*, Chapman and Hall, London, UK, 1980.

[12] B. Abraham and A. Chuang, "Outlier detection and time series modeling," *Technometrics*, vol. 31, pp. 241–248, 1989.

[13] M. Markou and S. Singh, "Novelty detection: a review—part 2: statistical approaches," *Signal Processing*, vol. 83, pp. 2499–2521, 2003.

[14] M. Markou and S. Singh, "Novelty detection: a review—part 1: statistical approaches," *Signal Processing*, vol. 83, pp. 2481–2497, 2003.

[15] R. Stephen and L. Arockiam, "Intrusion detection system to detect sinkhole attack on RPL protocol in Internet of Things," *International Journal of Electrical, Electronics and Computer Systems*, vol. 4, pp. 16–20, 2017.

[16] G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering," in *Proceedings of the 2016 3rd International Conference on Recent Advances in Information Technology (RAIT)*, vol. 7, Dhanbad, India, March 2007.

[17] J. Zhang and M. Zulkernine, "Anomaly based network intrusion detection with unsupervised outlier detection," in *Proceedings of the 2006 IEEE International Conference on Communications*, vol. 5, IEEE, Istanbul, Turkey, June 2006.

[18] G. Kaur, A. H. Lashkari, and A. Rahali, "Intrusion traffic detection and characterization using deep image learning," in *Proceedings of the 19th IEEE International Conference on Dependable, Autonomic & Secure Computing (DASC 2021)*, Calgary, Canada, June 2020.

[19] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, pp. 386–396, 2020.

[20] L. Zhu and N. Laptev, "Deep and confident prediction for time series at uber," in *Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, IEEE, New Orleans, LA, USA, November 2017.

[21] X. W. Chen and X. Lin, "Big data deep learning: challenges and perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014.

[22] V. Kindl, B. Skala, R. Pechanek, V. Kus, and J. Hornak, "Low-pass filter for HV partial discharge testing," *Sensors*, vol. 18, p. 482, 2018.

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

[24] J. Takeuchi and K. Yamanishi, "A unifying framework for detecting outliers and change points from time series," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, pp. 482–492, 2006.

[25] J. Zhu, Y. Wang, D. Zhou, and F. Gao, "Batch process modeling and monitoring with local outlier factor," *IEEE Transactions On Control Systems Technology*, vol. 27, pp. 1552–1565, 2019.

[26] L. E. Nugroho, L. Lazuardi, and A. S. Prabuwono, "Detection of anomalous vital sign of elderly using hybrid k-means clustering and isolation forest," in *Proceedings of the TENCON 2018-2018 IEEE Region 10 Conference*, IEEE, Jeju, Korea, October 2018.