Multi-task learning based Encoder-Decoder: A comprehensive detection and diagnosis system for multi-sensor data

Junfeng Wu1, Li Yao1, Bin Liu1, Zheyuan Ding1 and Lei Zhang2

Abstract
As more and more sensor data have been collected, automated detection, and diagnosis systems are urgently needed to lessen the increasing monitoring burden and reduce the risk of system faults. A plethora of researches have been done on anomaly detection, event detection, anomaly diagnosis respectively. However, none of current approaches can explore all these respects in one unified framework. In this work, a Multi-Task Learning based Encoder-Decoder (MTLED) which can simultaneously detect anomalies, diagnose anomalies, and detect events is proposed. In MTLED, feature matrix is introduced so that features are extracted for each time point and point-wise anomaly detection can be realized in an end-to-end way. Anomaly diagnosis and event detection share the same feature matrix with anomaly detection in the multi-task learning framework and also provide important information for system monitoring. To train such a comprehensive detection and diagnosis system, a large-scale multivariate time series dataset which contains anomalies of multiple types is generated with simulation tools. Extensive experiments on the synthetic dataset verify the effectiveness of MTLED and its multi-task learning framework, and the evaluation on a real-world dataset demonstrates that MTLED can be used in other application scenarios through transfer learning.

Keywords
Anomaly detection, anomaly diagnosis, event detection, multi-task learning, feature matrix

Date received: 30 September 2020; accepted: 29 March 2021

Handling Editor: James Baldwin

Introduction
Nowadays, diverse sensor data are increasingly collected and monitored in various real world applications, including spacecraft monitoring systems, medical diagnosis systems, protection of industrial assets, and etc. Most of these sensor data are in the form of time series and it is of great value to detect and diagnose potential threats in time series data to maintain the security and stability of various systems.

To monitor multi-sensor systems and give early warnings for operators, increasing approaches of anomaly detection, anomaly diagnosis, and event detection are proposed from different perspectives. Anomalies are represented by unusual or abnormal patterns in the data, which can be seen as the omen of faults. Some fault-related events which show no abnormal signs are also of interest to operators and important for system

1Science and Technology on Information System and Engineering Laboratory, National University of Defense Technology, Changsha, China
2Xi’an Satellite Control Center, Xi’an, China

Corresponding author:
Li Yao, Science and Technology on Information System and Engineering Laboratory, National University of Defense Technology, Changsha, Hunan 410073, China.
Email: 371865833@qq.com

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
monitoring and fault prevention.\textsuperscript{7} When the behavior of a dynamic phenomenon changes enough over time so as to be considered a qualitatively significant change, such a change is called an event.\textsuperscript{8} Obviously, an event is not necessarily an anomaly, and vice versa.

Anomaly detection approaches can be roughly divided into unsupervised ones and supervised ones. Unsupervised anomaly detection methods\textsuperscript{1,9–11} only use “normal” training samples to capture and model the normal behavior of a system. The learned model can be used to predict\textsuperscript{9} or reconstruct\textsuperscript{9–11} the input values or transformed features. When test data exhibit an unusual pattern, the prediction errors between the actual values and the predicted values are far larger than those of normal data, which indicates a potential anomaly. These unsupervised anomaly detection methods can theoretically detect all types of anomalies and be used in various applications without much domain priors, but they cannot provide detailed descriptions of detected anomalies since such information is not provided in the training phase. Besides that, such error-based approaches tend to find merely parts of anomaly sequences where they are most unpredictable, and thus they are also sensitive to noise data. Supervised anomaly detection approaches\textsuperscript{2,12} use labeled samples for training and they are more robust and superior in accuracy when the quality and quantity of the training samples are guaranteed.\textsuperscript{13,14} Besides that, they can provide additional information about the detected anomalies. However, in many real scenarios, it is difficult to obtain sufficient abnormal data for training, especially labeled ones. Moreover, these supervised anomaly detection approaches can hardly detect anomalies of unknown types.

Anomaly diagnosis aims to provide diagnosis information for operators to respond to the detected anomalies quickly. The research on anomaly diagnosis is often accompanied with that of anomaly detection, but not vice versa. In fact, most unsupervised anomaly detection approaches and systems only detect anomalies and leave the detected anomalies to additional diagnosis approaches or experts.\textsuperscript{15} which may delay the response to the anomalies. Only a few unsupervised algorithms\textsuperscript{9,10} consider anomaly detection and diagnosis in one framework. Most of the existing supervised anomaly detection approaches\textsuperscript{2,16} use classification models to classify the abnormal data into known anomalous classes. The obtained class labels represent the results of both anomaly detection and anomaly diagnosis. Current works regarding to anomaly diagnosis mainly pinpoint the anomalous components or classify the anomalies into known types, but more detailed information about the detected anomalies are not considered.

Event detection is widely used in various scenarios with different purposes, including multi-sensor system monitoring.\textsuperscript{7} Event detection approaches can also be divided into unsupervised ones and supervised ones. Unsupervised event detection\textsuperscript{17} can be used to detect usual patterns or anomaly events, while supervised event detection\textsuperscript{18} is mainly used to recognize fault-related events of known types. To the best of our knowledge, there is no work in which both anomalies and known fault-related events are considered for system monitoring or fault prevention.

To mitigate and balance the issues mentioned above, a novel scheme which can detect anomaly, diagnose anomalies and detect events simultaneously is presented. In this scheme, a Multi-Task Learning based Encoder-Decoder (MTLED) is proposed for anomaly detection, anomaly diagnosis and event detection. Multi-task learning\textsuperscript{19} aims at learning multiple related tasks in parallel and all of them can be implemented in real time due to its end-to-end architecture.

- Anomaly detection: to accurately detect abnormal parts in multivariate time series data.
- Anomaly diagnosis: to diagnose the anomalies and describe their abnormal information, including the types of anomalies, the severity of anomalies, the state of whether the anomalies end or not, and which channels are related to the occurrence of anomalies. This can help the system operators locate and handle incidences quickly. When the number of sensors in monitoring systems is large, this function is particularly helpful.
- Event detection: to detect some events which are related to known fault patterns, such as a sequence of sensor readings have exceeded a preset threshold for a long duration, or the average value of a sensor reading during a period is lower than a certain value. Such events are often used as important criteria for fault forecast and diagnosis, and thus also very important for system monitoring.

In MTLED, input data are fed in sliding windows and corresponding outputs are generated to achieve the above-mentioned functions in real time. MTLED is mainly composed of a task-shared encoder and several parallel decoders. The encoder is mainly composed of a convolutional neural network (CNN) and a Long Short-Term Memory networks (LSTM) module. The biggest difference between MTLED and traditional encoder-decoders\textsuperscript{20} is that the feature extracted by the
encoder retains the time dimension. Therefore, after encoding, a feature matrix is obtained rather than a feature vector. The main purpose of this design is to make the decoder of anomaly detection utilize the time dimension to achieve point-wise anomaly detection in an end-to-end way. In order to realize all the above tasks, a total of five decoders are designed. The first decoder is mainly an LSTM model, which is used for point-wise anomaly detection; the second decoder is a classification model, which classifies the type and severity of anomalies; the third decoder is used to discriminate whether anomalies end in the current sliding window; the fourth decoder is used to detect which channels are related to the occurrence of anomalies and the fifth decoder is used to detect events.

The main contributions of this study are as follows. Firstly, the shortcomings of the existing early warning systems are pointed out and analyzed systematically. Secondly, a novel schema which can detect anomalies, diagnose anomalies, and detect event simultaneously is proposed based on multi-task learning. Thirdly, in the proposed model, feature matrix is introduced into the encoder-decoder so that point-wise anomaly detection can be realized with an end-to-end deep learning model. Finally, we make public a large multivariate time series dataset which contains anomalies of multiple types, and experiments on the synthetic dataset and a real dataset demonstrate the effectiveness and generalizability of MTLED.

The rest of the paper is organized as follows. Section 2 presents the most relevant previous work. Section 3 introduces the preliminary works of MTLED and describes the architecture of MTLED as well as its training method. The experimental evaluation and discussions are articulated in Section 4. Conclusions and future directions of our work are given in Section 5.

Related work

The proposed MTLED aims to address the problems of anomaly detection, anomaly diagnosis, and event detection in a harmonized framework for fault prevention. In this section, the most related works regarding to these three parts are analyzed one by one.

Anomaly detection

Most of the unsupervised anomaly detection methods can detect anomalies at the granularity of time points. These point-wise anomaly detection methods usually take two main steps to detect anomalies. Firstly, the anomaly scores are calculated based on the prediction errors between the actual values and the predicted values for each time point. Then, a global threshold or a dynamic threshold is selected for anomaly scores to detect anomalies. A global threshold is usually learned with validation set which contains some labeled anomalous samples. Dynamic thresholds are set according to statistics in an unsupervised way. Both global thresholds and dynamic thresholds need to be determined by trial and error and cannot guarantee that the best threshold is selected. Different from these methods, the two-stage framework is replaced by an end to end deep learning model in MTLED, which can directly output point-wise results of anomaly detection. For this purpose, each point in sliding windows is classified into “normal” or “abnormal” class by feeding the point-wise features into a decoder whose main component is an LSTM model.

Supervised anomaly detection approaches usually extract features from time series and then classify the normal and abnormal class using feature vectors. A dual channel LSTM model is proposed by Zhao et al. to combine two types of features extracted from time data and force data for classification. Kim and Cho proposed a C-LSTM neural network which combines a CNN and RNN for automatic feature extraction and anomaly detection, where the output of this CNN layer is used as the input for several LSTM layers to reduce temporal variations. MALSTM-FCN combines FCN (fully convolutional network) and RNN features in a another way. Specifically, the state-of-the-art FCN module and LSTM module extract features in parallel and then the concatenated feature vector is passed onto a softmax layer for classification. The feature extraction part of MALSTM-FCN is similar to that of MTLED in that both of them concatenate the features from two parallel feature extraction modules. All the above supervised anomaly detection algorithms extract features and classify at the granularity of sliding windows. Thus, they can only detect anomalies at the granularity of sliding windows. In MTLED, features matrix is extracted without the loss of time dimension so that point-wise anomaly detection can be realized in a supervised learning framework. Besides that, the multi-task learning framework in MTLED can theoretically improve the ability of feature extraction and the performance of each task.

Anomaly diagnosis

Anomaly Diagnosis are divided into two subcategories: unsupervised approaches and supervised approaches. Most unsupervised anomaly detection approaches and systems are proposed to detect potential anomalies and leave the detected anomalies to experts, operators, or additional diagnosis tools. To reduce the response time, several unsupervised algorithms which consider anomaly detection and diagnosis in one framework are proposed. The anomaly diagnosis part merely gives the interpretations of the root causes. Zhang et al. attempt to solve the anomaly detection and diagnosis
using a multi-scale convolutional recurrent encoder-decoder. However, its anomaly diagnosis part just pinpoints the root causes according to top-k channels and simply assumes that the severity of an incident is proportional to the duration of an anomaly.

Most of the existing supervised anomaly detection approaches are mainly classification models which classify the abnormal data into known anomalous classes. The obtained class labels represent the results of both anomaly detection and anomaly diagnosis. Machine learning based classifiers are employed to detect and diagnose for cloud services, but only two types of known anomalies can be detected and diagnosed. Three kinds of neurodegenerative disease classes and one normal class are classified using a dual channel LSTM model. In our proposed model, more diagnosis information about the detected anomalies are provided by separate decoders, including the types of anomalies, the severity of anomalies, the state of whether the anomalies end or not, and which channels are related to the occurrence of anomalies. When anomalies occur, these diagnosis results can be obtained with anomaly detection results in the same time owing to the multi-task learning framework and this can greatly ease the burden of system operators.

Event detection

If the temperature of a boiler goes above a certain threshold, then it can be seen as an event. However, in some cases, the rule for determining whether sensor readings should generate an event is not that clear. Improvements in machine learning techniques, such as classification models, have accelerated the progress in event detection. For example, Batal et al. focuses the search on temporal patterns that are potentially more useful for classification and use support vector machine as classifiers. An LSTM model is used for Polyphonic Sound Event Detection. A sound event detection model which combines a convolutional bidirectional recurrent neural network with transfer learning is proposed.

In MTLED, event detection is accomplished by an FCN model whose input is the feature matrix shared by the task of anomaly detection and anomaly diagnosis. The feature matrix is trained on the data related to anomaly detection and anomaly diagnosis firstly, and shared as the input of the pre-trained model for event detection. Except the potential performance improvement, such a multi-task learning framework can theoretically alleviate the dependence on the amount of training samples for event detection.

MTLED

In this section, the preliminary works related to MTLED are introduced firstly. After that, the architecture of MTLED is presented. Finally, a three-stage training method for MTLED is described in detail.

Preliminary works

Temporal convolution. Temporal convolution is effective in feature extraction for time series classification problems, and 1-D convolution is widely used to extract temporal features. Given the univariate input signal \( x \in \mathbb{R}^L \) and the 1-D convolution kernel \( w \in \mathbb{R}^P \), the convolution output is \( y \in \mathbb{R}^{(L-P+1)} \), where \( L \) represents the length of the input sequence, \( P \) represents the length of the convolution kernel, and \((L-P+1)\) is the length of the output sequence. The output at time step \( m \) is calculated as follows:

\[
y^m = \sum_{p=0}^{P} f(w^p x^{(m+p)} + b)
\]

where \( w^p \) is the weight of the kernel at time point \( p \), \( x^{(m+p)} \) is the input at time point \( (m+p) \), \( b \) is bias, and \( f(\cdot) \) is the activation function.

As shown in Figure 1, the convolution result of input data is obtained by sliding convolution kernel with sliding stride \( s = 1 \). Generally speaking, in order to ensure the sufficiency of feature extraction, multiple convolution kernels are used for feature extraction. Consider a multivariable time series \( X \in \mathbb{R}^{L \times C} \), where \( L \) represents the length of the input sequence, \( C \) represents the number of channels, and \( N \) convolution kernel \( w_i \in \mathbb{R}^{P \times C} \) are used, the convolution output \( Z \in \mathbb{R}^{(L-P+1) \times N} \) is obtained.

LSTM. With the rapid development of deep learning, Recurrent Neural Networks (RNN), especially Long Short-Term Memory networks (LSTM), have been widely used for many sequence learning tasks, including speech recognition, natural language processing, and anomaly detection of time series. Compared with traditional RNN, LSTM improves the long-term memory by introducing a weighted self-loop which allows LSTM to forget past information in addition to

![Image](https://example.com/image.png)
accumulating it. The computation at each time step can be formulated as follows:

\[
\begin{align*}
    f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\
    g_t &= \sigma(W_c \cdot [h_{t-1}, x_t] + b_c) \\
    i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\
    C_t &= f_t C_{t-1} + i_t g_t \quad \text{(5)} \\
    o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\
    h_t &= o_t^* \tanh(C_t) \quad \text{(7)}
\end{align*}
\]

where \( \sigma \) denotes the logistic sigmoid function, \( * \) represents element-wise multiplication, \( W_f, W_c, W_i, \) and \( W_o \) are recurrent weight matrices, \( b_f, b_c, b_i, \) and \( b_o \) are bias, \( x_t \) is the input at time point \( t \), and \( h_t \) is the corresponding output.

**Attention mechanism.** In recent years, attention mechanism\(^{29} \) has been widely used in various fields of deep learning, including computer vision, speech recognition, and natural language processing. Attention mechanism originates from the study of human vision, and draws lessons from human attention mechanism. Due to the bottleneck of information processing, human beings will selectively pay attention to some information according to their experience while ignoring other information. From the physical sense, attention mechanism can be understood as the similarity measurement between current input and historical information. From the mathematical formula, attention mechanism can be understood as the weighted sum of input data according to the historical state and the characteristics of current input data.

The temporal attention computes the attention weights based on the previous hidden state and represents the input information as a weighted sum of the encoder hidden states across all the time steps.

The weighted sum of each input \( v' \) of each time step is used to compute the output as follows:

\[
\alpha' = \sum_{i=1}^{T} \alpha'_i v' \quad \text{(8)}
\]

The weight, \( \alpha'_i \), of each time step is calculated through

\[
\alpha'_i = \frac{\exp(\epsilon'_i)}{\sum_{j=1}^{L} \exp(\epsilon'_j)} \quad \text{(9)}
\]

Where the alignment model, \( \epsilon'_i \), is \( f_{att}(h^{(t-1)}, v') \), and \( h^{(t-1)} \) represents the historical information.

**The architecture of MTLED**

MTLED uses a shared encoder to obtain the feature matrix, based on which several task-related decoders are used to generate their own output, including results of anomaly detection, anomaly diagnosis and event detection. The MTLED network architecture is shown in Figure 2.

The overall framework of MTLED is a Multi-Task Learning based Encoder-Decoder. The encoder is mainly composed of a CNN and an LSTM module. Both modules extract features without losing the time dimension and their extracted features are concatenated along the time dimension. Then, a temporal attention mechanism is applied to select relevant features across all time steps and the feature matrix is obtained. In order to achieve all the above functions, five decoders are designed: one Decoder for Anomaly Detection (Dec_ADE), three Decoders for Anomaly Diagnosis (Dec_ADI), which provide anomaly types, anomaly states and anomaly channels respectively, and one Decoder for Event Detection (Dec_ED). The details of the encoder and decoders in MTLED are discussed below.

**Task-shared encoder.** A CNN and an LSTM module are used in the encoder for feature extraction. The CNN module consists of three stacked convolutional blocks. Each convolutional block consists of a one-dimension convolutional layer, which is accompanied by batch normalization\(^{30} \) followed by the ReLU activation function. The number of convolution kernels used for each convolutional layer are represented as \( N_1, N_2, \) and \( N_3 \). Specifically, 32 convolution kernels are used for each convolutional layer with filter sizes of 8, 5, and 3, respectively. To keep the time dimension unchanged, padding is applied for each convolutional layer. Consider a multivariate time series \( X \in R^{L \times C} \), where the length of the sequence is \( L \) and the number of the channels or variables is \( C \). After the three stacked convolutional blocks, the output of CNN module \( Z_1 \in R^{L \times N_1} \) is obtained, where \( N_3 = 32 \) in this case. The LSTM module consists of two LSTM layers, both of which contain 20 hidden units and are followed a dropout\(^{31} \) to combat overfitting. The output of LSTM module is \( Z_2 \in R^{L \times K_2} \), where \( K_2 = 20 \) denotes the number of the hidden units in the second LSTM layer. The feature matrix \( Z_1 \) and \( Z_2 \) are concatenated along the time dimension, then a temporal attention mechanism is applied to select relevant features across all time steps and the feature matrix \( Z \in R^{L \times F} \) is obtained, where \( F = N_3 + K_2 \). Figure 3 presents how the dimensions change to concatenate the features extracted from the above two modules in the encoder, in which the layers that will not cause a dimension change are removed for clarity.
As can be seen from Figure 3, both feature extraction modules retain the time dimension in the whole process, so that their features can be concatenated and still retain the features for each time point. This is the basis of the point-wise anomaly detection in our model.

Decoder for anomaly detection. Different from error-based anomaly detection methods which need to select thresholds for anomaly score sequences, MTLED classifies each time point into “normal” or “anomalous” class directly. All the anomalous time points are labeled 1, while all the other normal points are labeled 0. Dec_ADE is mainly an LSTM followed by a dropout to output the initial anomaly detection results. LSTM is chosen due to its memory and the continuity of the output in the time dimension to prevent a continuous anomaly sequence being split into too many segments.

To accelerate the training phase and regularize the anomaly detection results, the initial anomaly detection results are concatenated with the final outputs of the last sliding window as the auxiliary input, and a dense (fully connected) layer with “sigmoid” as activation function is used for the final anomaly detection result of the current window. Given that the input feature matrix is $Z^t \in \mathbb{R}^{N \times F}$, the dropout output of LSTM is $y^{t_0} \in \mathbb{R}^L$, the final output of the last sliding window is $y^{t-1} \in \mathbb{R}^L$, the concatenated vector $[y^{t_0}, y^{t-1}]$ is used for the classification to obtain the final anomaly detection result $y^f \in \mathbb{R}^k$. As Figure 4 shows, the prediction of $y^f$ for each time point is saved as historical data and used as auxiliary input for its next prediction.

Decoder for anomaly diagnosis. All the three decoders for anomaly diagnosis (Dec_ADI) are classifiers. Since multiple anomaly types and anomaly channels are allowed in one sliding window, multi-label classifiers are applied for anomaly type and anomaly channel. On the contrary, only one anomaly state is defined for each

Figure 2. Network architecture of MTLED.

Figure 3. The illustration of dimensions change in the encoder.
sliding window, and thus a multi-class classifier is more appropriate and adopted. These decoders share a similar structure, that is, a global average pooling (GAP) layer followed by a dense layer. The multi-label classifiers use “sigmoid” as activation function while the multi-class classifier uses “softmax” (see Figure 2).

The decoder for classifying anomaly types can recognize four typical anomaly types and detect other anomaly types. The typical anomaly types include “point,” “shift,” “trend,” “variance,” whose definitions are given in the next section. The severity of these typical anomaly types is classified into two groups “subtle” and “severe.” For example, a “severe point” anomaly deviate from their adjacent values more noticeably than a “subtle point” anomaly. All the other anomaly types except the four typical types are merged into one class to reinforce the generalization ability of the decoder so that anomalies of unknown types can also be detected. Thus, nine classes of anomaly types are considered and predicted by the decoder: subtle point, severe point, subtle shift, severe shift, subtle variance, severe variance, subtle trend, severe trend, and others.

Obviously, an on-going anomaly is more threatening than an ended one. The decoder for anomaly state analysis is used to distinguish whether anomalies end at the end of each sliding window. These two kinds of states, represented by 0 and 1, can be easily discriminated and important for system monitoring.

The decoder for pinpointing the root causes or the anomaly channels is used to locate the anomaly channels which are related to the anomaly parts in the current sliding window. Only the channels closely related to the anomalies are denoted as 1 while others are denoted as 0.

**Decoder for event detection.** Unlike anomalies, events related to corresponding known faults are usually defined according to different domains, thus the event detection part can hardly be shared across different applications. To make the event detection part have a wider application prospect, it is important to alleviate its dependence on labeled data for training. A decoder for event detection (Dec_ED) relying merely on a small amount of training samples is designed.

First of all, several data augmentation strategies for time series are used to increase the number of training samples. Secondly, the training of decoder makes use of the encoder pre-trained on the anomaly detection and diagnosis parts. Lastly, an FCN model whose input is the feature matric is used for further feature extraction and classification. The FCN consists of a convolutional block and a GAP layer followed by a sigmoid dense layer for classification (see Figure 2).

**The training of MTLED**

The training phase needs two training sets Tr₁ and Tr₂. Tr₁ contains sufficient multivariate anomalous samples with anomaly annotation, and it is to train the sections of MTLED which are related to anomaly detection and anomaly diagnosis. Tr₂ contains a few multivariate samples with event labels and is used to train the sections of MTLED which are related to event detection.

There are three stages in the whole training process of MTLED. In the first training stage, the encoder, Dec_ADI and Dec_ADE without the auxiliary input are trained, that is, the sections of Dec_ED and the auxiliary input in Dec_ADE are ignored in this stage. After that, the model M₀ is obtained, and the initial predicted output of Dec_ADE and Dec_ADI with M₀ can be obtained.

In the second stage, the initial predicted output obtained from Dec_ADE with M₀ is fed into the Dec_ADE as auxiliary input to regularize the output as shown in Figure 4. After stage 2, the model M₁ is obtained and the final predicted output of Dec_ADE and Dec_ADI can be obtained with M₁.

In the last stage, data augmentation is used for Tr₂ to mitigate the problem of small sample size of training data for event detection. Some commonly use strategies, including cropping, jittering, and time warping,³³ are applied to generate Tr₂'. The training of Dec_ED makes use of the encoder pre-trained on Dec_ADE and Dec_ADI by fine-tuning on Tr₂'.

Since multi-task learning is applied in MTLED, the loss for each task is defined firstly. Among the five decoders we use, only the one for anomaly state is a multi-class classifier, while the other four are multi-label classifiers. For the multi-class classifier, a “categorical cross entropy” loss is defined as follows:

\[
L(y, \hat{y}) = - \sum_{i=1}^{n} \sum_{j=1}^{m} y_{i,j} \log \hat{y}_{i,j} \tag{10}
\]
Where $n$ is the number of training samples for the multi-class classifier, and $m$ is the number of classes. For the multi-label classifiers, a “binary cross entropy” loss is defined as follows:

$$L(y, \hat{y}) = -\sum_{i=1}^{o} \sum_{j=1}^{p} (\hat{y}_{i,j} \log y_{i,j} + (1-\hat{y}_{i,j}) \log (1-y_{i,j}))$$

where $o$ is the number of training samples for the multi-label classifiers, and $p$ is the number of classes.

For simplicity, all the decoders are given the same weight. In the first two stages of training, the sum of loss for Dec_ADE and Dec_ADI is optimized, and the loss for Dec_ED is optimized in the last training stage. Mini-batch stochastic gradient descent together with the Adam optimizer is used to minimize the loss function.

**Experimental evaluation**

MTLED needs to be trained in a supervised manner. To this end, the synthetic dataset MTSD_AMT which contains all kinds of annotation labels needed is introduced and MTLED is evaluated on MTSD_AMT. After that, MTLED is evaluated on a real-world dataset through transfer learning.

**MTSD_AMT**

To the best of our knowledge, there is no existing public dataset which contains enough annotated anomaly samples to enable a thorough empirical evaluation for the proposed MTLED. Thus, a large-scale multivariate time series dataset which contains anomalies of multiple types is generated with the help of the simulation tools (Agots-master and DeepADoTS-master). These simulation tools take the correlation of variables into consideration and all the labels needed can be acquired easily without expert priors. In Agots-master, four typical sensor-level anomaly types are defined: “point,” “shift,” “variance,” “trend.” Point anomalies are single values that noticeably deviate from their adjacent values, shift anomalies are sequences that noticeably deviate from their surrounding sequences, variance anomalies are sequences that oscillates within a large range, and trend anomalies are sequences which show an obvious trend of ascending or descending. The severity of the generated anomalies can be controlled by the parameter in the simulation program. Besides that, other anomaly types such as the sinusoidal waves with white noises, and other harmonic signals are generated by DeepADoTS-master, and all the other anomalies except the above four typical types are labeled as “other” class.

Figure 5(a) shows an example of multivariate time series including a point anomaly (x4), a variance anomaly (x3) and another anomaly like a sinusoidal wave (x5). In Figure 5(b), an example of a trend anomaly (x2) and a shift anomaly (x1) is shown.

Each of the four typical anomalies is further classified into two classes according to its severity. Thus, in our annotation, a total of one normal class and nine classes of anomaly type labels are given: subtle point, severe point, subtle shift, severe shift, subtle variance, severe variance, subtle trend, severe trend, and others.

Two events are defined manually for the experiments regarding to event detection. Event 1 occurs when the minimal reading of second sensor in a duration (lasting more than 120 time points) exceeds the...
threshold 0.75. Event 2 occurs when the average value of third sensor in a duration (lasting more than 150 time points) is lower than −0.5. Only a few samples are provided for training the sections related to event detection to demonstrate that it is feasible to apply MTLED in various scenarios where abundant annotated data are not always available.

Table 1 describes the brief information of MTSD_AMT. MTSD_AMT is available on https://github.com/jlro001/MTSD_AMT.

Table 1. The brief information of MTSD_AMT.

|                      | Training | Test  |
|----------------------|----------|-------|
| # of channels        | 6        | 6     |
| # of total samples   | 5000     | 5000  |
| # of sensor-level anomalies | 8738   | 8745  |
| Maximum sequence length | 2000   | 2000  |
| Anomaly duration ratio | 14.22  | 14.18 |
| # of samples for event detection | 80     | 5000  |

**Experiment setup**

The length of the input sequence equals to the size of the sliding window and is set as 200 provided a balance between performance and training time. The stride of the sliding window for training and test is set to 10. We set the batch size as 128 for training, and run for 200 epochs with early stopping and an initial learning rate of 0.001 which decays every 10 epochs. All the dropout rates in MTLED are set to 0.3. A magnitude scale is specified first and is used to normalize the input multivariable time series.

Point-wise evaluation metrics are used to compare MTLED with unsupervised anomaly detection methods. The number of true positives (TPp) are the number of predicted “anomalous” points which belong to “anomalous” sequences. The number of false positives (FPp) are the number of predicted “anomalous” points which do not belong to “anomalous” sequences. The number of false negatives (FNp) are the number of predicted “normal” points which belong to “anomalous” sequences.

For supervised anomaly detection methods which do not tell the exact part of anomalies, we take a sliding window as a sample. A true positive is recorded if a predicted “anomalous” sliding window contain any true labeled sequence, a false positive is recorded if a predicted “anomalous” sliding window contain no true labeled sequence, and a false negative represents a predicted “normal” sliding window which actually contains any true labeled sequence. The number of these true positives is denoted as TPpS, the number of false negatives is denoted as FNpS, and the number of false positives is denoted as FPpS. The performance evaluation on anomaly diagnosis and event detection also takes each sliding window as a sample. The metrics used for evaluation include the following:

- **Precision (P)**: \( P = \frac{TP}{TP + FP} \);
- **Recall (R)**: \( R = \frac{TP}{TP + FN} \);
- **F-measure (F)**: \( F = 2 \times P \times R / (P + R) \).

F-measure considers Precision and Recall simultaneously without any bias, and it is used to evaluate the overall performance. \( P_p, R_p, \) and \( F_p \) represent the metrics used in the point-wise evaluation, while \( P_s, R_s, \) and \( F_s \) evaluate the performance with a sliding window as a basic unit. All the experiments in this work are repeated five times and the average results are reported for a fair comparison.

**Evaluation on MTSD_AMT**

The performances of anomaly detection, anomaly diagnosis, and event detection are evaluated on MTSD_AMT. After that, several variants of MTLED are compared to analyze the internal components of MTLED.

**Anomaly detection result.** To demonstrate the effectiveness of MTLED, we first compare it with several state-of-the-art unsupervised anomaly detection approaches. These baseline methods are listed as follows.

- **EncDec-AD**: An LSTM based encoder-decoder is used to reconstruct the input sequence, and the resulting reconstruction errors are modeled as a multivariate Gaussian distribution and is used to detect anomalies.
- **LSTM-NDT**: A stacked LSTM model is used to predict the future values using the current
sequences, and the prediction errors are smoothed using exponential weighted moving average (EWMA), based on which dynamic thresholds according to statistic information are selected to detect anomalies.

- OmniAnomaly: VAE and GRU (a variant of RNN) are integrated in a novel way such that the temporal dependence and stochasticity of time series can be explicitly modeled. The reconstruction probabilities and the threshold selected according to the principle of extreme value theory are used to detect anomalies.

All these unsupervised models are firstly trained on normal samples before the anomalies are injected in MTSD_AMT. LSTM-NDT and OmniAnomaly can determine thresholds without any positive samples, while EncDec-AD select the threshold based upon the training on anomaly samples.

As can be seen from Figure 6, MTLED outperforms all the baseline methods significantly when point-wise metric is applied for anomaly detection. Among the baseline methods, LSTM-NDT and OmniAnomaly get similar results. They have lower recall compared with EncDec-AD, and it is mainly because EncDec-AD can utilize the anomaly information in MTSD_AMT. We argue that these error-based methods tend to find just parts of anomaly sequences where they are most unpredictable, while MTLED can treat each point of the anomaly sequence equally. Besides that, error-based methods are relatively sensitive to noise data, which leads to more false positives. In general, the multi-task learning framework based model shows an obvious advantage over the methods which merely concentrate on anomaly detection.

Supervised anomaly detection methods usually take each sliding window as a sample and use classification models for anomaly detection. As argued in Fawaz et al., FCN achieved the best performance for time series classification among nine commonly used MTS classification methods. LSTM is the most successful variant of RNN, and is also widely used as a baseline in supervised MTS classification. C-LSTM combines a CNN and RNN for automatic feature extraction and anomaly detection, in which the output of this CNN layer is used as the input for several LSTM layers. Another state of the art MTS classification model, MALSTM-FCN, is proposed in Karim et al. MALSTM-FCN concatenates feature vectors from a fully convolutional block and an LSTM block and uses a softmax layer for the final classification.

The above methods are compared with MTLED and their results are presented in Figure 7. Among all these methods, although MTLED achieves the highest precision, its overall performance ($F$) is still lower than MALSTM-FCN because of its lower recall. In general, MALSTM-FCN show a slight better performance than other methods including MTLED. This may attribute to the extra squeeze and excite block used in MALSTM-FCN. Although MTLED did not achieve the best performance among the supervised methods, the performance difference between MTLED and MALSTM-FCN is very little. More importantly, MTLED can detect the exact part of anomalies, while these supervised methods can only classify sliding windows as “abnormal” or “normal.” The great improvement of MTLED on point-wise anomaly detection can overshadow its ordinary performance among these supervised methods.

### Anomaly diagnosis result

Three respects of anomaly diagnosis are included: anomaly type, anomaly state, and anomaly channel. Since all these diagnosis tasks are realized by classification models, the baseline models used for supervised anomaly detection comparison are also used for evaluating the performance of anomaly diagnosis. In the three diagnosis tasks, anomaly type classification is more challenging than the other two tasks because the severity of the anomalies is also considered in the anomaly type classification for the four typical anomalies.

Table 2 reports the performance comparison of each respect of anomaly diagnosis where the best scores are highlighted in bold-face. In terms of precision, different methods achieve the highest score on each task and no one is significantly better than others. MALSTM-FCN seems to show a slight advantage with respect to recall ($0.6958$ and $0.9791$) in the tasks of classifying anomaly type and anomaly state. But MTLED beats MALSTM-FCN on recall ($0.8838$ vs $0.8623$) on the task of detecting which channels are related to the occurrence of anomalies. In summary, MALSTM-FCN achieves the best $F$-measure in terms of anomaly type, while MTLED achieves the best $F$-measure in the other two tasks. Even though the conclusion that MTLED is superior to all the baseline methods cannot be drawn,
MTLED at least shows a comparable performance with the state-of-the-art anomaly diagnosis methods.

**Event detection result.** The event detection performance of MTLED and the state-of-the-art classification models are shown in Figure 8. For a fair comparison, data augment is used for all these methods.

From Figure 8, we can easily find that all the methods show a similar result on precision, while MTLED shows a relatively significant improvement on recall than all the other methods, and thus obtaining a better $F$-measure. Different from other baseline methods, MTLED utilizes the model pre-trained on the tasks of anomaly detection and anomaly diagnosis, and this might be the main reason that MTLED shows a better performance on the task of event detection and validates the effectiveness of the multi-task learning framework in MTLED.

**Internal analysis of MTLED.** Feature matrix is important in our model because that is the basis of the point-wise anomaly detection in an end-to-end way. In this subsection, we mainly discuss the function of the auxiliary input in Dec_ADE, the temporal attention mechanism and the multi-task learning framework. We denote MTLED without auxiliary input as MTLED_aux, MTLED without attention layer as MTLED_att, and the model without auxiliary input and attention layer as MTLED_aux_att. The model which ignores the sections of other tasks other than anomaly detection is denoted as MTLED_MTL. For a fair comparison, all the other parts of each models except those mentioned above are exactly the same.

As shown in Figure 9, the best results in terms of all metrics are achieved by MTLED. In general, the overall performance of MTLED is improved when the attention layer, the auxiliary input, and multi-task learning are applied. More specifically, MTLED_aux_att performs the worst in terms of precision, and both MTLED_aux and MTLED_att have a lower precision than MTLED. It indicates that both the attention layer and the auxiliary input can improve the precision of MTLED. MTLED_MTL achieves the lowest recall, which demonstrates that other related tasks, such as anomaly diagnosis and event detection, contribute much to improve the recall of anomaly detection. This result further verifies the effectiveness of the multi-task learning framework in MTLED.

**Evaluation on space shuttle**

To validate that MTLED can also be used for real scenarios, comparison experiments are also conducted on a real-world dataset. Space Shuttle contains the telemetry data about the Energize/De-Energize cycle of space shuttles. TEK 14 and TEK 17 are used in this experiment and both samples contain 5 periodic sequences with 1000 points per cycle. The train-test split is

---

**Table 2.** Anomaly diagnosis result of MTLED and other baseline approaches.

| Anomaly type | Anomaly state | Anomaly channel |
|--------------|---------------|-----------------|
|              | $P_s$ | $R_s$ | $F_s$ | $P_s$ | $R_s$ | $F_s$ | $P_s$ | $R_s$ | $F_s$ |
| FCN          | 0.8712 | 0.6315 | 0.7322 | 0.9850 | 0.9717 | 0.9783 | 0.9429 | 0.8262 | 0.8807 |
| LSTM         | 0.8782 | 0.6252 | 0.7304 | 0.9805 | 0.9738 | 0.9771 | 0.9392 | 0.6058 | 0.7365 |
| C-LSTM       | 0.8451 | 0.6586 | 0.7403 | 0.9914 | 0.9703 | 0.9808 | 0.9554 | 0.8347 | 0.8910 |
| MALSTM-FCN   | 0.8295 | 0.6958 | 0.7568 | 0.9877 | 0.9791 | 0.9833 | 0.9645 | 0.8623 | 0.9106 |
| MTLED        | 0.8510 | 0.6738 | 0.7521 | 0.9904 | 0.9782 | 0.9843 | 0.9629 | 0.8838 | 0.9217 |

The best performance indicators are displayed in bold.
consistent with that of the work of others. Only “normal” samples are included in the training set and the amount of training data is relatively limited. To solve the problem of class imbalance and data insufficient in the training set, only five training epochs are used to fine-tune the MTLED model pre-trained on MTSD_AMT to find a balance between precision and recall. The telemetry values about the normal part of the dataset have strong periodicity, but the anomalies are very different and the change of values is relatively subtle. Space Shuttle is more challenging than MTSD_AMT and is mainly used to further verify the effectiveness and generalizability of MTLED.

Since the number of channels in Space Shuttle is less than that of MTSD_AMT, zero padding is used to adjust the input data to the model pre-trained on MTSD_AMT. The dataset contains no ground-truth annotation about anomaly diagnosis or event detection, thus we merely report the anomaly detection performance of MTLED and other baseline methods. Because EncDec-AD provides no method for choosing anomaly score threshold when no anomaly sample is available for training, all possible anomaly score thresholds are enumerated to search for the best F-measure, which is presented in Figure 10.

As can be seen from Figure 10, MTLED shows a significantly higher recall than all the other baseline methods, and thus obtain a better overall performance. Even the best F-measure of EncDec-AD is still lower than the F-measure of MTLED. This is reasonable in that it is much easier for MTLED to detect more parts of the anomalies than other baseline methods which confront the problem of decreasing the precision when improving the recall. On the other hand, the anomalies in Space Shuttle are relatively subtle, which makes it more difficult to select a proper threshold for error-based anomaly detection methods. Such a result validates that MTLED pre-trained on MTSD_AMT can also be used for real scenarios effectively through fine-tuning on a few samples.

**Conclusions**

In this paper, the shortcomings of mainstream detection and diagnosis methods for multi-sensor data are summarized and analyzed from their own perspectives. In summary, the single task, such as anomaly detection, anomaly diagnosis, or event detection, can hardly satisfies all the needs in system monitoring and fault prevention. In order to solve these issues, MTLED which can detect anomaly, diagnose anomalies and detect events simultaneously is proposed in a multi-task learning scheme. A large-scale multivariate time series dataset MTSD_AMT with sufficient annotation information is generated to enable an adequate supervised pre-training of our proposed model.

In MTLED, feature matrix is introduced into an encoder-decoder framework for the purpose of extracting features at the granularity of time point so that point-wise anomaly detection can be realized naturally. The encouraging results of point-wise anomaly detection of MTLED demonstrate the value of the feature matrix. In the meantime, other decoders based on the same feature matrix is used for anomaly diagnosis and event detection. Experiments on MTSD_AMT show that MTLED shows a significant advantage on anomaly detection, a comparable performance on anomaly diagnosis and a slight improvement on event detection. Furthermore, internal analysis on the variants of MTLED verifies the respective functions of its multi-task learning framework, attention mechanism and the auxiliary inputs. On the real dataset Space Shuttle, the superiority of MTLED on anomaly detection is validated and it also demonstrates that MTLED can be used for various domains by fine-tuning on small-scale samples.

MTLED provides an intuitive and effective way to realize various tasks related to system monitoring and early warning in one deep learning model. As the cost of the function perfection and performance improvement, the training of MTLED is more complicated and it needs to take more computational overhead than trivial models mainly due to the feature matrix. While the training phase is more time-consuming than most approaches, MTLED can still be used for monitoring the multi-sensor data in real-time. There is much research works can be done to optimize the proposed model, such as the addition of squeeze and excite block for improved feature extraction. Reducing the computational overhead is also very important so that such a detection and diagnosis system can be deployed in various application scenarios.
Author contributions

The individual contributions of the authors are as follows: conceptualization, J.W., L.Y., and L.Z.; methodology, J.W., L.Y., L.Z., and B.L.; software, J.W. and Z.D.; formal analysis, J.W., L.Y., and B.L.; investigation, J.W., L.Z., and Z.D.; writing—original draft preparation, J.W.; writing—review and editing, J.W., L.Y. and B.L.; supervision, L.Y.; project administration, B.L.; funding acquisition, L.Y.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The presented work is framed within the Research Project under Grant GK2019A010296. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

ORCID iD

Junfeng Wu https://orcid.org/0000-0001-5848-2805

Notes

1. https://github.com/KDD-OpenSource/agots
2. https://github.com/KDD-OpenSource/deepadots
3. https://github.com/chickenbestlover/RNN-Time-series-Anomaly-Detection

References

1. Hundman K, Constantinou V, Laporte C, et al. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, London, UK, 19–23 August 2018, pp.387–395. ACM.
2. Zhao A, Qi L, Dong J, et al. Dual channel LSTM based multi-feature extraction in gait for diagnosis of Neurodegenerative diseases. Knowl Based Syst 2018; 145: 91–97.
3. Wielgosz M, Skozen A and De Matteis E. Protection of superconducting industrial machinery using RNN-based anomaly detection for implementation in smart sensor. Sensors 2018; 18: 3933.
4. Nanduri A and Sherry L. Anomaly detection in aircraft data using Recursive Neural Networks (RNN). In: 2016 Integrated Communications Navigation and Surveillance (ICNS), Herndon, VA, 19–21 April 2016, pp.5C2-1–5C2-8. New York, NY: IEEE.
5. Tanaka S, Nakada K, Okita T, et al. Anomaly diagnosis of sensors for linear discrete dynamical system based on analytical redundancy. Trans Soc Instrum Control Eng 2009; 21: 525–527.
6. Liu S, Che H, Smith K, et al. Contamination event detection using multiple types of conventional water quality sensors in source water. Environ Sci Process Impacts 2014; 16: 2028–2038.
7. Davoodi M, Meskin N and Khorasani K. Event-triggered multiobjective control and fault diagnosis: a unified framework. IEEE Trans Industr Inform 2017; 13: 298–311.
8. Guralnik V and Srivastava J. Event detection from time series data. In: knowledge discovery and data mining, San Diego, CA, 15–18 August 1999, pp.33–42.
9. Zhang C, Song D, Chen Y, et al. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In: National conference on artificial intelligence, 2019, pp.1409–1416.
10. Su Y, Zhao Y, Niu C, et al. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, Anchorage, Alaska, 4–8 August 2019, pp.2828–2837.
11. Malhotra P, Ramakrishnan A, Anand G, et al. LSTM-based encoder-decoder for multi-sensor anomaly detection. arXiv: Artificial Intelligence. 2016.
12. Kim TY and Cho S. Web traffic anomaly detection using C-LSTM neural networks. Expert Syst Appl 2018; 106: 66–76.
13. Liu D, Zhao Y, Xu H, et al. Opprentice: towards practical and automatic anomaly detection through machine learning. In: Internet measurement conference, Tokyo, Japan, 28–30 October 2015, pp.211–224.
14. Shipmon DT, Gurevitch JM, Piselli PM, et al. Time series anomaly detection: detection of anomalous drops with limited features and sparse examples in noisy periodic data. arXiv: Machine Learning. 2017.
15. Han QZ and Yong Y. A wavelet-based approach to abrupt fault detection and diagnosis of sensors. IEEE Trans Instrum Meas 2001; 50: 1389–1396.
16. Sauvanaud C, Kaaniche M, Kanoun K, et al. Anomaly detection and diagnosis for cloud services: practical experiments and lessons learned. J Syst Softw 2018; 139: 84–106.
17. Duan K and Zhang W. Event-triggered fault-tolerant control for networked systems with dynamic quantiser. IET Control Theor Appl 2016; 10: 1088–1096.
18. Kim Y and Park W. A study on cyber threat prediction based on intrusion detection event for APT attack detection. Multimed Tools Appl 2014; 71: 685–698.
19. Ranjan R, Patel VM and Chellappa R. HyperFace: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Trans Pattern Anal Mach Intell 2019; 41: 121–135.
20. Liu X, Liu Y, Zhang M, et al. Improving stockline detection of radar sensor array systems in blast furnaces using a novel encoder–decoder architecture. Sensors 2019; 19: 3470.
21. Karim F, Majumdar S, Darabi H, et al. Multivariate LSTM-FCNs for time series classification. Neural Netw 2019; 116: 237–245.
22. Long J, Shelhamer E and Darrell T. Fully convolutional networks for semantic segmentation. In: Computer vision and pattern recognition, Boston, MA, 7–12 June 2015, pp.3431–3440.
23. Batal I, Fradkin D, Harrison JH, et al. Mining recent temporal patterns for event detection in multivariate time series data. In: Knowledge discovery and data mining, Beijing, China, 12–16 August 2012, pp.280–288.
24. Hayashi T, Watanabe S, Toda T, et al. Duration-controlled LSTM for polyphonic sound event detection. IEEE Trans Audio Speech Lang Process 2017; 25: 2059–2070.
25. Jung S, Park J and Lee SW. Polyphonic sound event detection using convolutional bidirectional LSTM and synthetic data-based transfer learning. In: International conference on acoustics speech and signal processing, Brighton, UK, 12–17 May 2019, pp.885–889.
26. Wang Z, Yan W and Oates T. Time series classification from scratch with deep neural networks: a strong baseline. In: International joint conference on neural network, Anchorage, AK, 14–19 May 2017, pp.1578–1585.
27. Bontemps L, McDermott J and Le-Khac N-A. Collective anomaly detection based on long short-term memory recurrent neural networks. In: International conference on future data and security engineering, Can Tho City, Vietnam, 23–25 November 2016, pp.141–152. Springer.
28. Park D, Hoshi Y and Kemp CC. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. IEEE Robot Autom Lett 2018; 3: 1544–1551.
29. Yan C, Tu Y, Wang X, et al. STAT: spatial-temporal attention mechanism for video captioning. IEEE Trans Multimedia 2020; 22: 229–241.
30. Ioffe S and Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv: Learning. 2015.
31. Srivastava N, Hinton GE, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014; 15: 1929–1958.
32. Tsoumakas G and Katakis I. Multi-label classification: an overview. Int J Data Warehouse Min 2007; 3: 1–13.
33. Wen T and Keyes R. Time series anomaly detection using convolutional neural networks and transfer learning. arXiv: Learning. 2019.
34. Kingma DP and Ba J. Adam: a method for stochastic optimization. arXiv: Learning. 2014.
35. Hunter JS. The exponentially weighted moving average. J Qual Technol 1986; 18: 203–210.
36. Fawaz HI, Forestier G, Weber J, et al. Deep learning for time series classification: a review. Data Min Knowl Discovery 2019; 33: 917–963.
37. Saleh K, Hossny M and Nahavandi S. Driving behavior classification based on sensor data fusion using LSTM recurrent neural networks. In: International conference on intelligent transportation systems, Yokohama, Japan, 16–19 October 2017.
38. Hu J, Shen L and Sun G. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, 18–22 June 2018, pp.7132–7141.