Machine learning-based climate time series anomaly detection using convolutional neural networks

R. Srinivasan[1], L. Wang[1] and J.L. Bulleid[1]

[1]*National Institute of Water and Atmospheric Research (NIWA), Private Bag 14901, Wellington, New Zealand*

Correspondence: Raghav.Srinivasan@niwa.co.nz , +64-4-386-0525

Key words: machine learning, deep learning, climate data quality control, image recognition, convolutional neural networks, time series anomaly detection, chart mining

Abstract

Data from New Zealand's National Climate Network are operationally verified both during data ingest and post data ingestion into the National Climate Database. The quality control process in the database verifies the data in two ways: automated checks, such as where data are automatically checked for 'out-of-expected-range' values, or for consistency with nearby site data (buddy checks), and another approach where a quality control analyst manually scrutinises data for anomalies. In the latter approach, climate timeseries data from sensor networks are plotted for quality control purposes and these plots are manually analysed for anomalies by a quality control analyst on a weekly basis. This manual process is performed in addition to other manual and automated quality checks because it helps to identify additional anomalies or unusual patterns in data that were not caught by the automated quality control processes. As more observational capacity is added to the climate network, manually reviewing quality plots becomes increasingly time-consuming, cumbersome and costly, and has an increasing potential to compromise data quality. In this study, we explore an image-based method to automate the manual anomaly detection process on quality control plots using deep learning. To do this we trained a Convolutional Neural Network (CNN) model with images of time series quality plots. The model learned to identify plots that contained anomalies similar to those a manual reviewer would detect. We have successfully achieved a high anomaly classification score using a CNN with modified VGG-16 architecture. We were also able to successfully identify, and colour-highlight, the classified anomalous regions within the quality plots using Gradient-based Class Activation Mapping. We achieved an overall anomaly classification F1 score of 0.92 and anomaly localisation accuracy of 91%.

1. Introduction

1.1 Background

New Zealand's National Climate Database, hosted by the National Institute of Weather and Atmospheric Research (NIWA), stores observational data from climate stations located across the country. These include stations managed by NIWA, MetService, Regional Councils, Fire and Emergency New Zealand and other agencies. The

Submitted: 21 July 2020
Accepted: 3 November 2020

data serves a multitude of applications, from weather and hazard forecasting, to scientific and economic studies and commercial activities. Around 1.8 billion rows of data are stored for extraction from the database, for both public and private use. There are about 2000 regular users and 50000 registered database users who have extracted data at some time in the past decade. Curation and quality control of these data are key to ensuring the data are fit for purpose for a multitude of uses.

NIWA operates the National Climate Network (NCN) that comprises hundreds of climate monitoring stations. Data telemetered from each station within the NCN are quality checked as part of the process of ingestion into the National Climate Database (CliDB). Quality control (QC) checks are performed on these datasets using both automated and manual verification. As part of the manual QC process, data from the NCN are plotted to produce raw weekly timeseries QC plots (Figure 1) and these plots are used for manual data verification. When verified, the data are securely archived in CliDB. These QC plots are produced by individual climate stations and there are hundreds of QC plots produced every week that require manual review.

1.2 Study aims

In this study, we aim to augment the manual review process by applying a deep learning solution to automatically recognise, and flag potentially erroneous data. We treated this as an image classification task and used a Convolutional Neural Network (CNN) to classify and flag data anomalies to the attention of a QC analyst. This helps minimise the manual effort otherwise needed

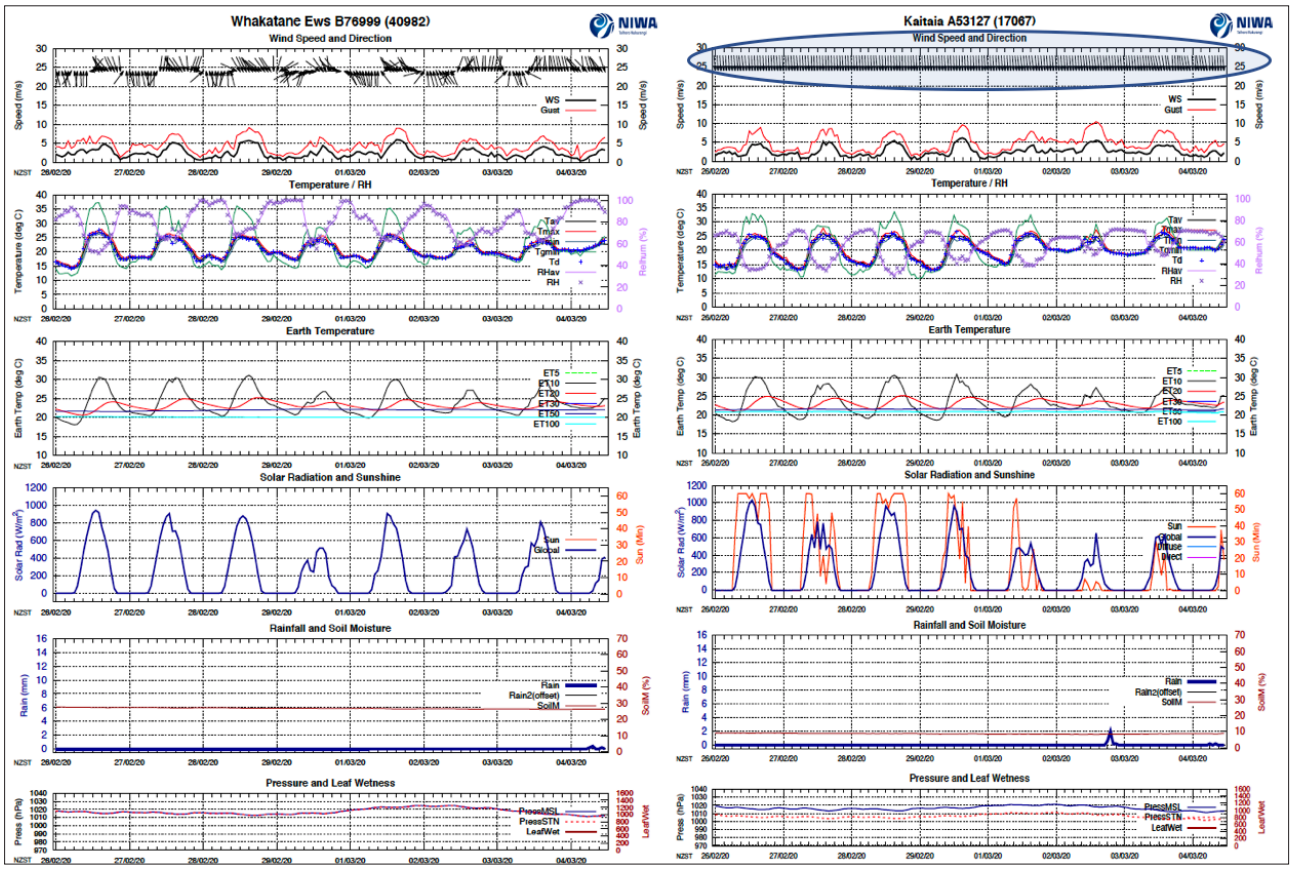


Figure 1: Two examples of weekly QC plots produced from individual climate stations before verification and archival. Left: a plot showing no pattern of anomalies. Right: a plot that should be flagged as anomalous because the wind direction sensor (circled) has been stuck in the same position for the whole week.

to review all the plots and reduces the manual task to one of remediating the flagged anomalies.

It is common to use a Recurrent Neural Network (RNN) model in time series analysis problems due to their temporal/sequential nature (Hundman et al., 2018; Park et al., 2018; Karim et al., 2018; Hüsken et al., 2003). In our study, we were trying to reproduce a manual QC review process whereby a person looks at the image of a quality plot and detects anomalous features within it. We treated this as an image classification problem because we intended to replicate the manual review process. Figure 2 illustrates the current QC plots review process and we aimed to retain this overall QC process by replacing the manual component of the review with an automated process. In addition, these images were already operationally produced, and an archive of these plots were readily available to train and test the algorithm. We decided to use a CNN for detecting anomalies as the CNN model was suited to image classification tasks.

There has been some previous work done where CNNs are used on time series images. Zheng et al. (2014); Yang et al. (2015); Cui et al. (2016) successfully used a deep CNN on time series data for image classification. Also, Zhao et al. (2017) similarly successfully used their multi scale convolutional neural network (MCNN) for time series classification tasks. In particular, Wen et al. (2019) developed a transfer learning-based CNN framework where U-net (Ronneberger et al., 2015) inspired CNN was used successfully for anomalous image segmentation in time series data.

In the above-mentioned CNN related works, the input signal was fed into the CNN either through transformations on the sensor signal or as a one-dimensional input of the sensor signal itself. In our study, the input was the 2-dimensional image of the QC plots and the aim was to perform chart mining on these images to detect anomalies. There has been some earlier work done

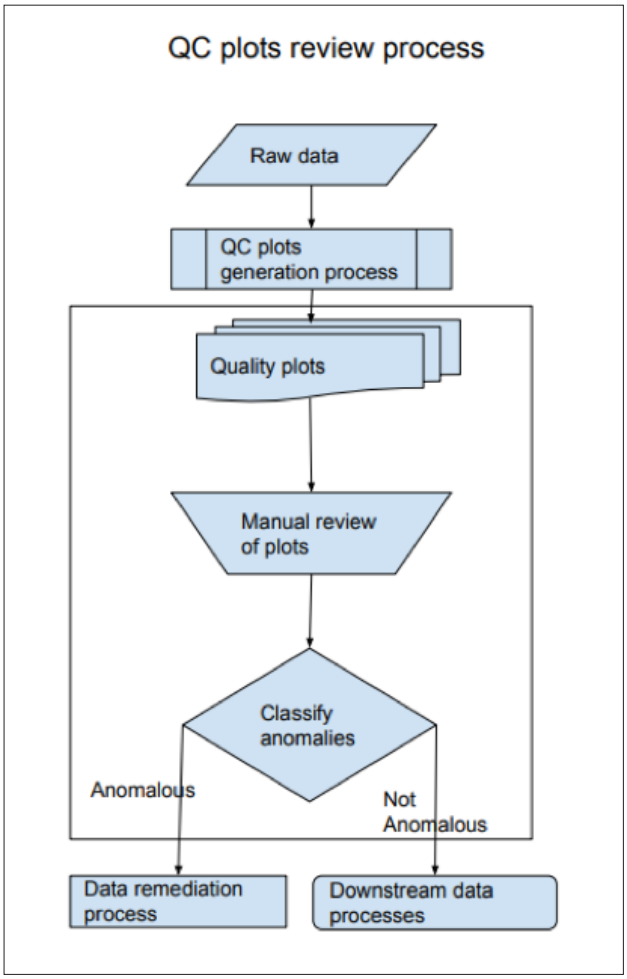


Figure 2: A high-level overview of the QC plots review process. The processes inside the box above could be changed to an automated CNN-based image classification task.

on chart mining (Davila et al., 2020). One past study used CNNs for chart type classification and extracted data, and text, from charts using image processing techniques and Optical Character Recognition (OCR) respectively (Balaji et al., 2018). CNNs were used to detect text or numerical elements in images of the charts along with their type (Liu et al., 2019, Cliche et al., 2017). To the best of our knowledge, our work is the first attempt in detecting anomalous patterns from charts using CNNs on sensor data for QC purposes.

We aimed to use the VGG-16 based CNN architecture for the purposes of this study. VGG-16 (Simonyon et al., 2014) network architecture was developed by the Visual

Geometry Group (VGG), Department of Computer Science, University of Oxford. Their architecture consists of 16 weighted layers (dubbed VGG-16). This approach replaced the large filter sizes of previous CNN architectures, such as AlexNet (Krizhevsky et al., 2012), with small 3 pixel x 3 pixel filters. VGG-16 uses an input image size of 224 pixels x 224 pixels x 3 channels (RGB). The network consists of five convolutional blocks with max pooling at the end of each convolutional block. The first two blocks have two convolutional layers each. The next three blocks have three convolutional layers each (Figure 3). The VGG-16 network was trained and tested on the ImageNet (Deng et al., 2009) dataset, comprising more than 14 million images and 1000 different classes. VGG-16 has a good reputation having secured first place in the ILSVRC - ImageNet Large Scale Visual Recognition Competition (Russakovsky et al., 2014) for low localisation error, and won second place for low classification error.

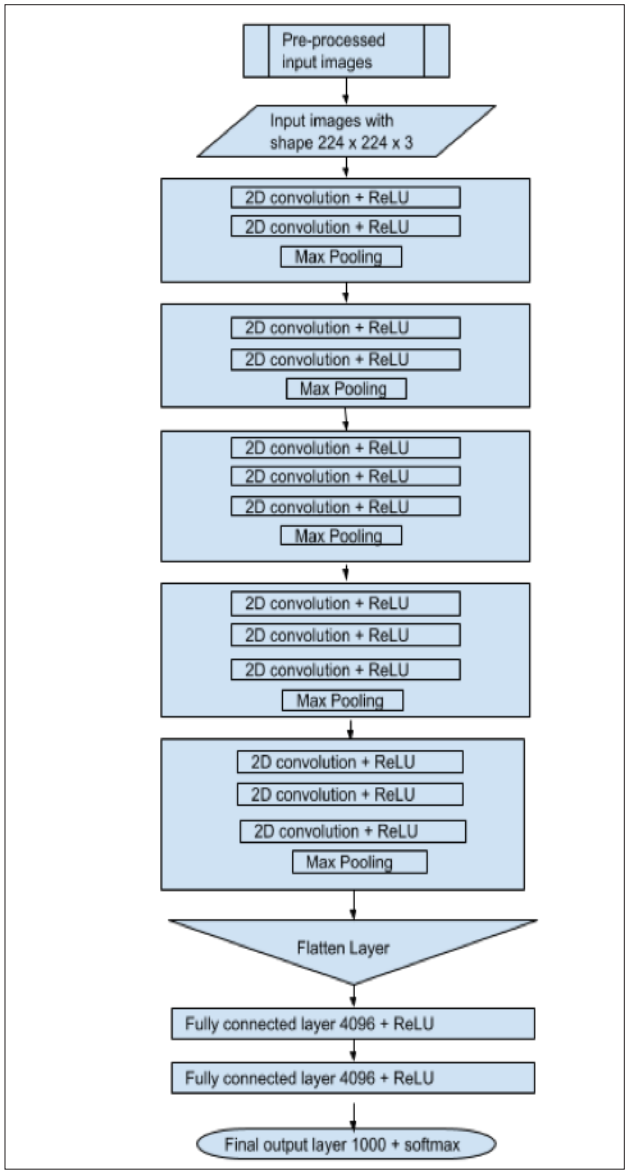


Figure 3: VGG16 architecture diagram that shows the five convolutional blocks along with its two fully-connected and softmax output layers.

2. Train/test data selection

This Section explains the QC plots, how the training and testing datasets for the model were chosen, and describes the pre-processing steps involved before feeding the training images into the model.

2.1 QC plots

Most of the data archived into CliDB are ingested in near-real time. Some of the QC checks are performed during data ingestion and some checks are performed as a batch process, post-ingestion. As explained in Section 1.1, an aspect of NIWA's CliDB QC process involves generation of timeseries plots for each station, on both a daily and weekly schedule (developed by NIWA Climate Database Technician Errol Lewthwaite). These plots either contain hourly data, or data at 10-minute intervals if available. These QC plots are generated from the pre-ingest data and this helps to check if the routine data ingest QC processes have correctly identified the quality issues. Currently, these QC plots are manually reviewed, post data ingestion, to ensure the quality of the archived data. This manual review of plots is one of the many aspects of manual quality control. The QC plots capture many basic climate variables in a single plot, for each station. Figure 1 shows the following variables: wind direction, wind speed, wind gust, maximum temperature, minimum temperature, mean temperature, grass minimum temperature, relative humidity (RH), earth temperature profile (5 cm, 10 cm, 20cm, 30 cm, 50 m, 100 cm), sunshine, solar radiation,

rainfall, soil moisture, leaf wetness and mean barometric pressure at sea-level. Representing all of these variables on every plot enables the analyst to explore any inter-dependency and thereby add 'context' to clarify observations that facilitate QC decisions. For example, a sudden upward spike in soil moisture could be related to a rainfall event; an increase or decrease in air temperature could be related to a change in wind direction. Also, there are subtle relationships between solar radiation and sunshine and between wind and RH. Since these inter-dependencies play an important role in QC decision making, we have decided to use the entire plot to train our model, hence encompassing all the variables and their format. Not all stations measure all of the above climate variables. Hence, while the format is the same for all stations, some variables may be absent.

2.2 Data selection

The QC plot generation process has been used for several years and weekly QC plots spanning the last six years are archived in the CliDB database system. In our study we have used these archived plots to identify various scenarios and flag the corresponding images. We manually reviewed these plots and chose around 1000 (both valid and anomalous) to train the model. We included plots over different seasons and from different monitoring stations.

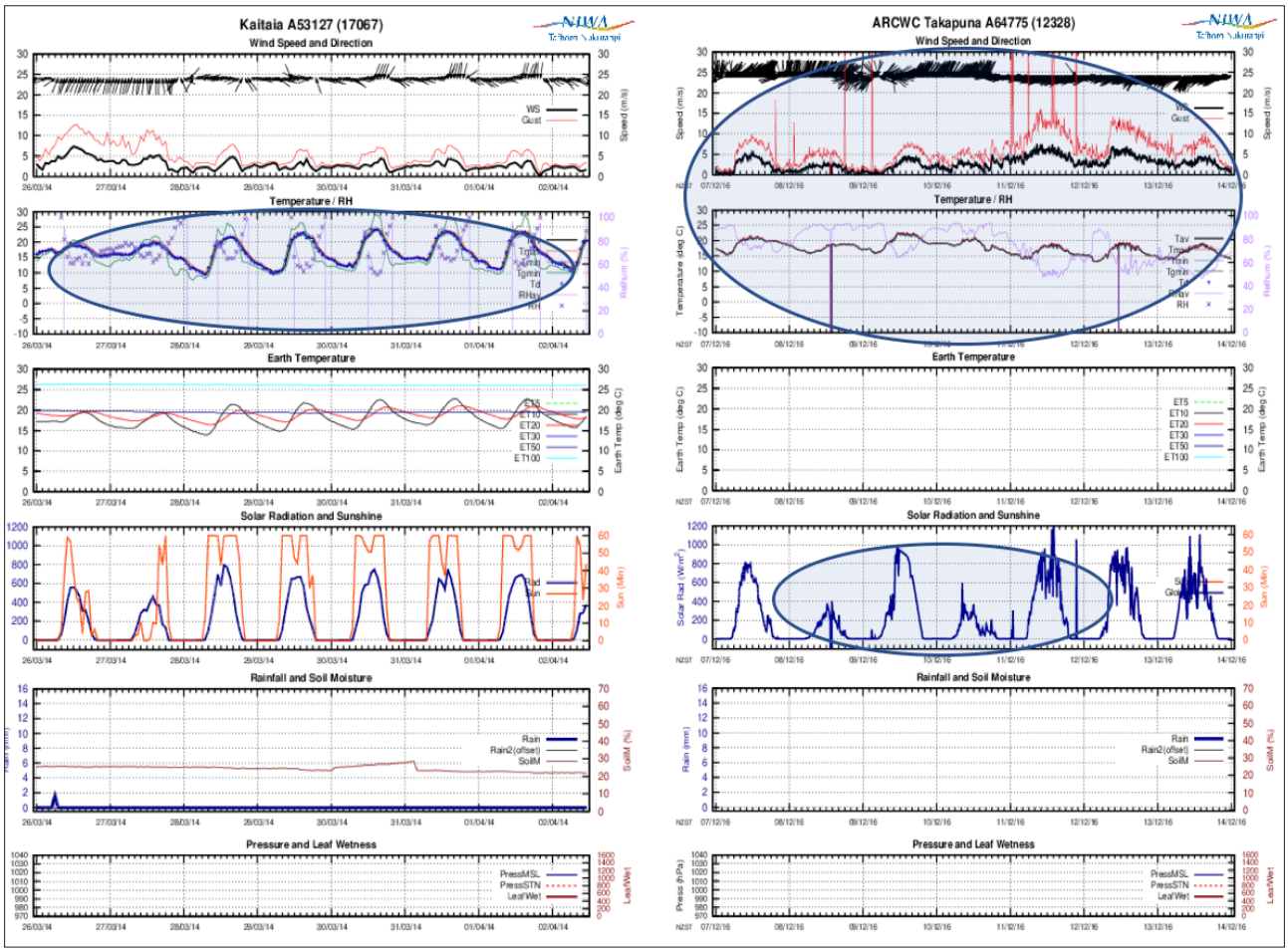


Figure 4: Left: the relative humidity (RH) time series shows sudden drops to zero; Right: there was an issue with sudden peaks in wind gust to beyond the displayed axis scale, and sudden drops in RH, temperature and radiation values (which may be deliberate default settings so that missing values can be recognised).

We chose plots representing various anomaly features for different variables. These anomalies are characterised by their patterns and broadly defined as:

a. Sudden peaks/drops;

b. Stuck values;

c. Uncorrelated behaviour of related variables;

d. Unusual pattern/fluctuation;

e. Incomplete timeseries.

a. Sudden peaks/drops

A large sudden drop can occur when there is a missing value in a timeseries, and the instrument channel defaults to reporting a pre-set extreme low value. A large sudden peak could be a value that is out of range or within a valid range but resulting from some other issue. These sudden peaks or drops could be due to a range of issues such as instrument calibration, communication, or power supply faults. The QC plots are generated from this pre-ingested data to highlight these issues (Figure 4). The data ingest process checks the data for minimum and maximum range before ingesting into CliDB. If there is an out of range value observed, the observation will be assigned 'suspect' on ingest into the climate database. These range thresholds are set for each site and parameter observed.

Also, a sudden peak may sometimes be valid if, for example, it is the result of an extreme weather event. In some cases of extreme values, we would classify those data as valid if the correlated variables were also displaying similar patterns.

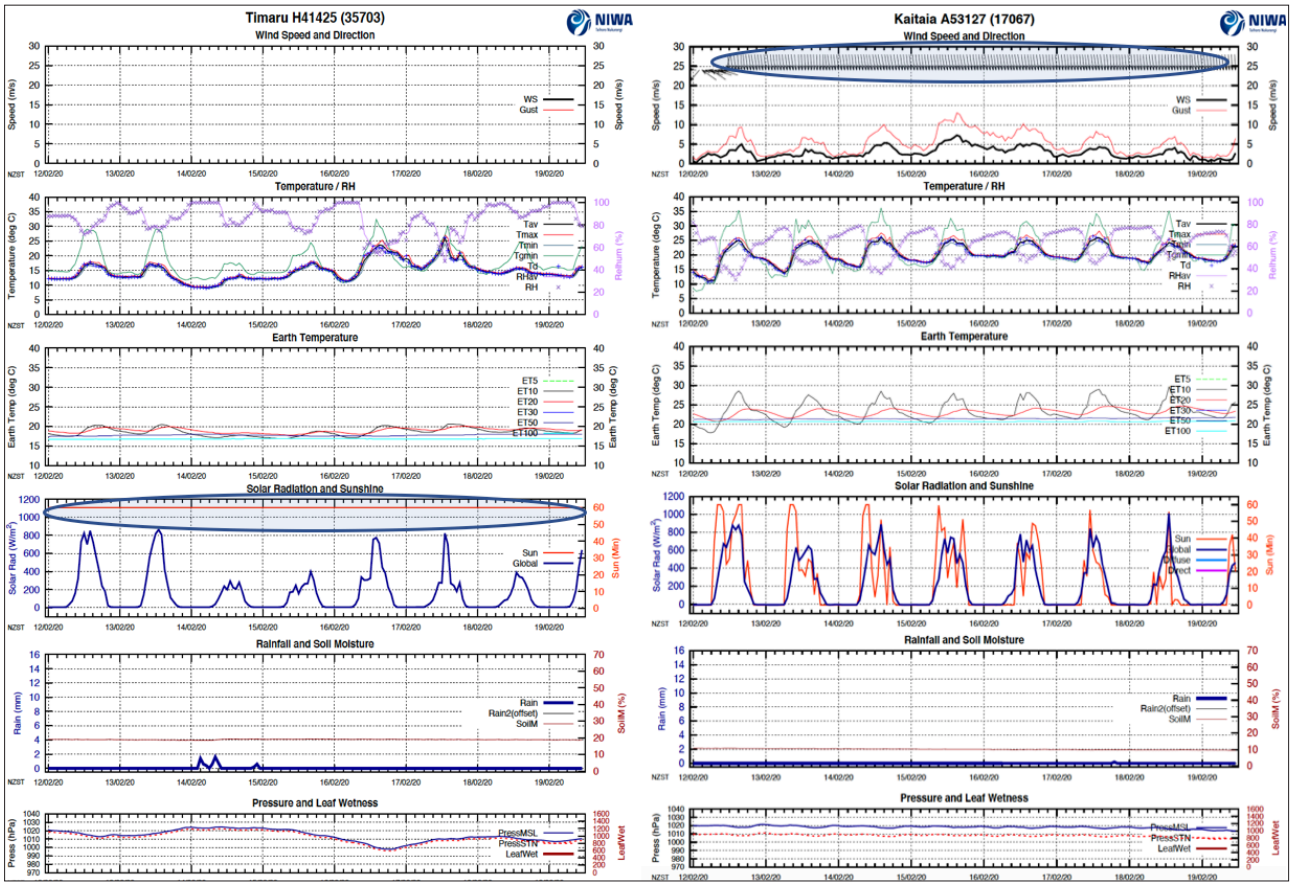


Figure 5: Left: sunshine is stuck, as indicated by the straight line in the timeseries; Right: wind direction appears to be stuck in one direction.

b. Stuck values

Stuck values may be indicated by a straight line or an unchanging/repetitive pattern. It could be stuck for a short duration within the timeseries, or for the whole plot period. Except for wind direction, remaining variables' stuck values could be identified by a horizontal line. Wind direction stuck values were identified by wind barb vectors pointing in the same direction (Figure 5). In other variables the lines could appear horizontal, vertical or slanted. Periods of zero rainfall and radiation lie within valid ranges, so are not interpreted as stuck values.

c. Uncorrelated behaviour of related variables

This anomaly relates to dissimilar behaviour of a certain variable with respect to its correlated variable(s). For example (Figure 6), an increase in soil moisture, without a corresponding rainfall event, should be flagged for further investigation. Also, there was an example where the 10 cm earth temperature was consistently higher than its corresponding 5 cm, 20 cm, 30 cm, 50 cm, 100 cm temperatures. We expect the 10 cm earth temperature to be closer to the 5 cm and 20 cm values.

d. Unusual fluctuations/blips

This issue relates to unusual patterns or fluctuations in the timeseries. These need not be extreme but could result from measurement errors, potentially caused by faulty instruments, even when values fall within the defined acceptable range. For example, Figure 7 shows a site whose 20 cm earth temperature is perturbed by small blips that occur too frequently and too briefly to be a natural environmental event.

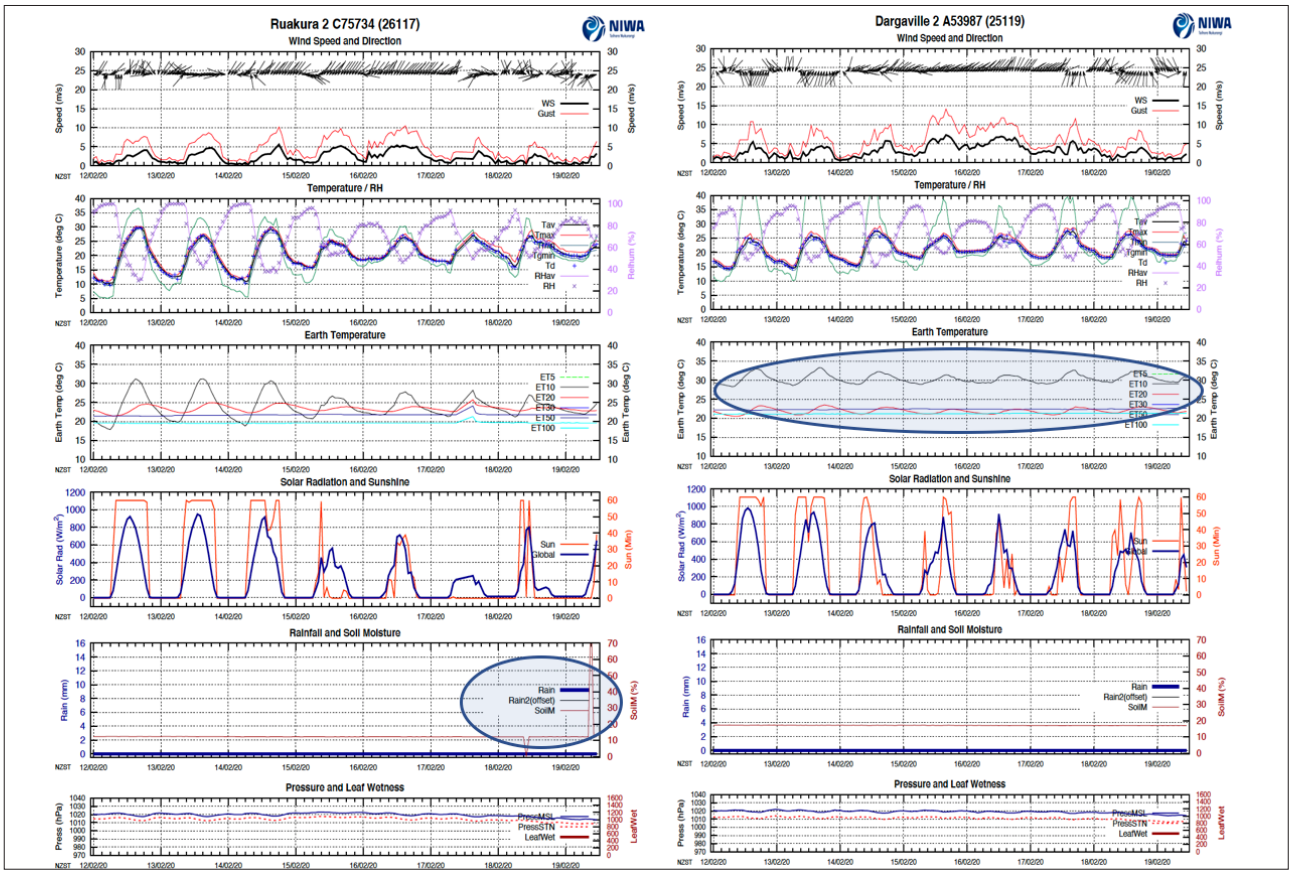


Figure 6: Left: a sudden increase in soil moisture value without a corresponding rainfall event; Right: the 10 cm earth temperature time series differed abnormally from the corresponding 5 cm and 20 cm values.

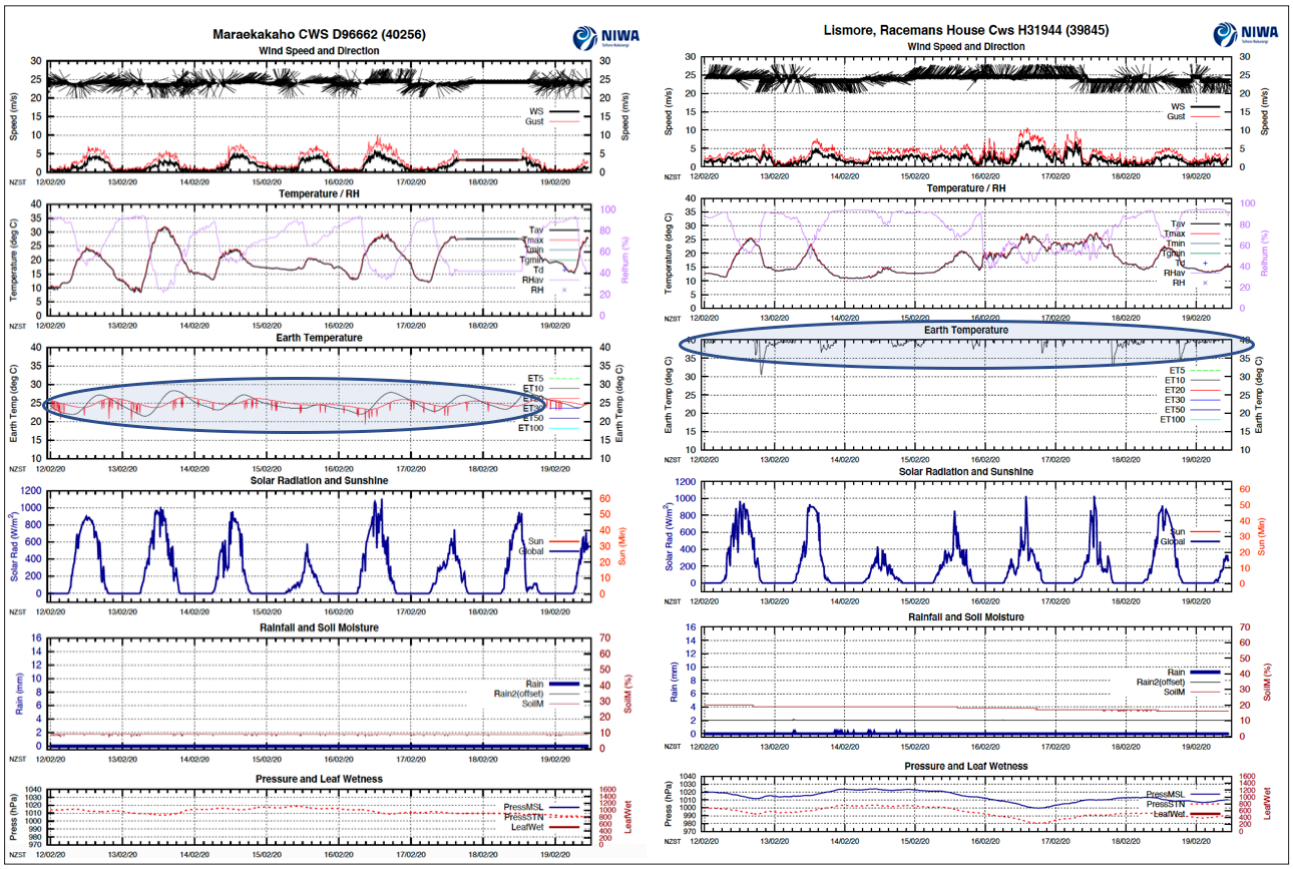


Figure 7: Left: the 20 cm earth temperature (red line) had small and frequent blips indicating a faulty instrument. Right: earth temperature is abnormally high and has an unusual pattern.

e. Incomplete timeseries

If an instrument stops measuring during the plot period, the QC plots would display gaps, or sometimes a plot might rescale the time axis to fit only the available observed values. Gaps between measurements in a particular time series are typically indicated by prescribed error default values. These are set during the data ingest process, to aid easy recognition of missing data in cases where remaining sensors are reporting normally. We classified both categories of missing data as incomplete timeseries (Figure 8).

2.3 Pre-processing

In order to meet input requirements for the VGG-16 based model, we converted the archived pdf quality plots to PNG. During conversion, we applied lossless compression in our plots, to retain any uniquely representative features that could affect classification accuracy. The resultant size of the PNG images was 792 x 612 x 3 pixels. We used this image size as the input to our VGG-16 based model. The PNG files produced were 8-bit images. This included 8-bit RGB channels and an 8-bit alpha channel. The images we used for training were rendered within the sRGB colour profile. We used ImageMagick v6.9.9-26 Q16 x86_64 in conjunction with Ghostscript v9.07 to generate the PNG files. In addition, we normalised the pixel values by dividing by 255. The images were presented with plot axis labels and the station name for training with the assumption that the algorithm would either learn with the labels or learn to ignore them. The Y axis scales in the plot images are not always constant, as the scale changes with season and station location (e.g. NZ North Island, NZ South Island and Antarctica).

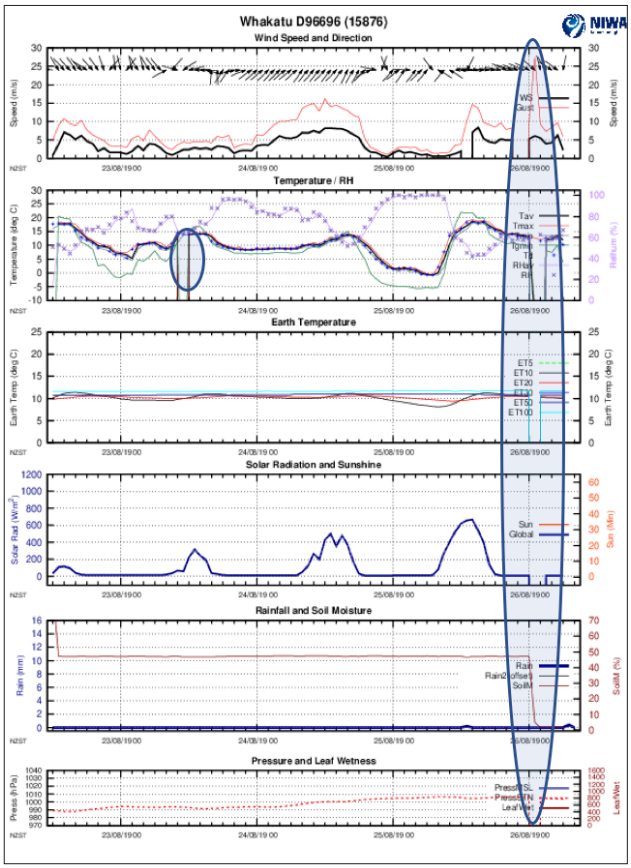


Figure 8: This station reported only four days of data for the whole week, and there were also missing values within the reported time series during the four days.

3. Network architecture, transfer learning and results

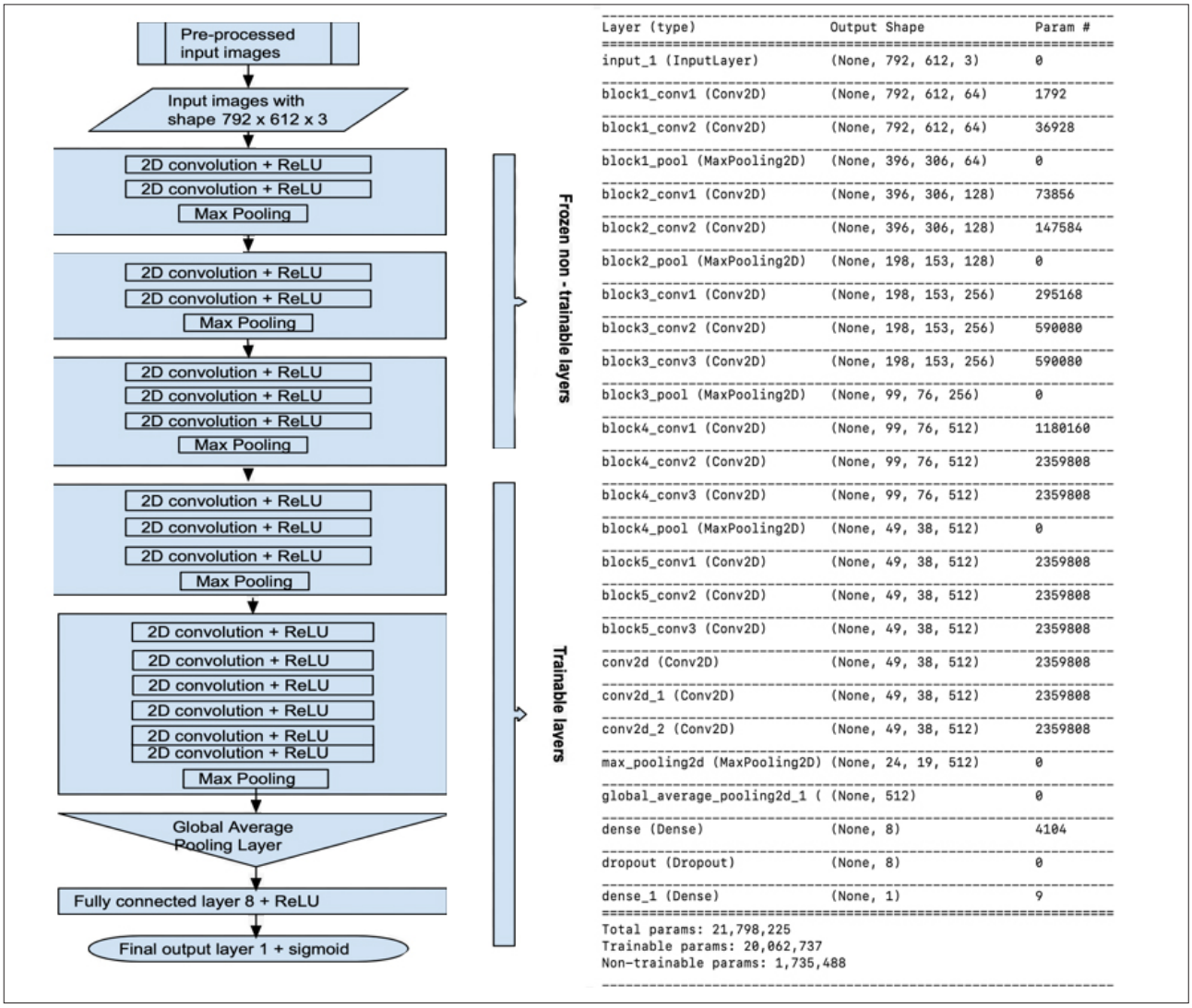
In this section we describe, in detail, the network architecture that we have used for this study, along with the results of tests done on our validation dataset. Also, we discuss our anomaly localisation process, using Gradient-based Class Activation Mapping (Selvaraju et al., 2016).

3.1 Network architecture

We use the 2-dimensional QC chart images as an input into CNN for anomaly classification. As described in Section 1.2, VGG-16 has a good reputation for low classification and localisation errors. Since we are interested in both anomaly classification and localisation, we chose VGG-16 (Figure 3) network architecture for our study. This architecture is known for its simplicity and accuracy. We have used transfer learning, which is a method of fully, or partially, applying weights (knowledge) from an existing model that has been trained on a large set of images for a different problem, and customising it to solve a related problem (West et al., 2007).

We have modified the last convolutional block of VGG-16 to include three more convolutional layers with 512 filters because it improved our training and testing loss values. The filter size for these layers was 3 x 3 pixels. We have replaced the flatten layer with a global average pooling (GAP) layer (Lin et al., 2013) as it helped to minimise overfitting. Lin et al. (2013) in their study used a GAP layer instead of the flatten layer and fully connected layer, and they were able to minimise overfitting. We chose GAP as we thought that it would perform well in extracting the feature maps due to the large amount of white space in our images. On top of the GAP layer, we have added a fully connected layer of eight neurons with ReLU activation (Agarap et al., 2018) followed by a dropout and the final output layer with a sigmoid activation function. We have used Stochastic Gradient Descent (SGD) (Bottou et al., 2018) as the optimizer and binary cross-entropy loss function for training this model. Figure 9 represents the network architecture that was used in this study. Keras coding library was used for the purposes of this study (Chollet et al., 2015).

As discussed in Section 2.3, our input image size was set at 792 x 612 x 3 pixels. In our study, we have frozen the weights from VGG-16 that was trained on the ImageNet dataset for the first nine layers in the architecture (Figure 9) and retrained the remaining layers. We tried different frozen and trainable layer combinations and this combination yielded the best accuracy during our training process.



Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 792, 612, 3)	0
block1_conv1 (Conv2D)	(None, 792, 612, 64)	1792
block1_conv2 (Conv2D)	(None, 792, 612, 64)	36928
block1_pool (MaxPooling2D)	(None, 396, 306, 64)	0
block2_conv1 (Conv2D)	(None, 396, 306, 128)	73856
block2_conv2 (Conv2D)	(None, 396, 306, 128)	147584
block2_pool (MaxPooling2D)	(None, 198, 153, 128)	0
block3_conv1 (Conv2D)	(None, 198, 153, 256)	295168
block3_conv2 (Conv2D)	(None, 198, 153, 256)	590080
block3_conv3 (Conv2D)	(None, 198, 153, 256)	590080
block3_pool (MaxPooling2D)	(None, 99, 76, 256)	0
block4_conv1 (Conv2D)	(None, 99, 76, 512)	1180160
block4_conv2 (Conv2D)	(None, 99, 76, 512)	2359808
block4_conv3 (Conv2D)	(None, 99, 76, 512)	2359808
block4_pool (MaxPooling2D)	(None, 49, 38, 512)	0
block5_conv1 (Conv2D)	(None, 49, 38, 512)	2359808
block5_conv2 (Conv2D)	(None, 49, 38, 512)	2359808
block5_conv3 (Conv2D)	(None, 49, 38, 512)	2359808
conv2d (Conv2D)	(None, 49, 38, 512)	2359808
conv2d_1 (Conv2D)	(None, 49, 38, 512)	2359808
conv2d_2 (Conv2D)	(None, 49, 38, 512)	2359808
max_pooling2d (MaxPooling2D)	(None, 24, 19, 512)	0
global_average_pooling2d_1 (	(None, 512)	0
dense (Dense)	(None, 8)	4104
dropout (Dropout)	(None, 8)	0
dense_1 (Dense)	(None, 1)	9

Total params: 21,798,225
Trainable params: 20,062,737
Non-trainable params: 1,735,488

Figure 9: Left: The VGG-16 network-based architecture with modifications to its last block and the fully connected layers. The first three blocks were frozen and existing VGG-16 ImageNet weights were used. The remaining layers were trained. Right: Model summary of different layers with corresponding shapes.

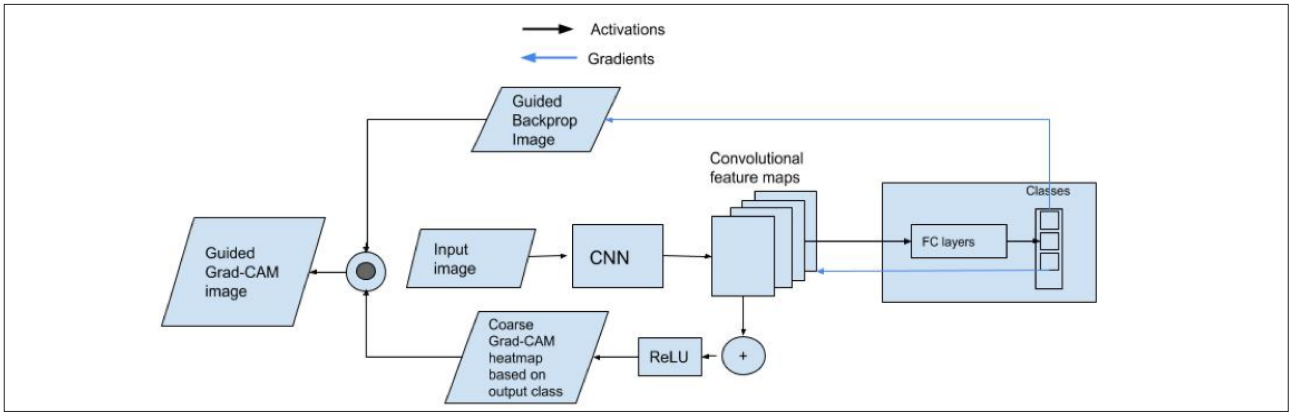


Figure 10: An overview of the Grad-CAM process (Selvaraju et al., 2016). As can be seen, the gradients that are backpropagated based on the output class are combined with the convolutional feature maps to generate the coarse Grad-CAM localisation. After which, this heatmap is pointwise multiplied with the guided backpropagation image to get the final Guided Grad-CAM visualisation.

3.2 Anomaly localisation using Gradient based Class Activation Mapping

In this section, we discuss our anomaly localisation process. In addition to classifying anomalous images, we aimed to identify the anomalous regions within the plots. We considered the class activation map (CAM) based technique for anomaly localisation.

Zhou et al. (2015) demonstrated in their study using their CAM technique that high accuracy classification and object localisation could be achieved using a GAP layer without training on any bounding box annotation. In that study, they performed GAP on the last convolutional layer that outputs the spatial average of the feature map and uses these as features for the final output fully-connected layer. Based on this, the important regions of the image can be identified by projecting back the weights of the class in the output layer to the convolutional feature maps of the last convolutional layer. This highlights the regions of the images based on the final classification derived by the network. But this technique could be applied to a convolutional network with no fully connected layers and where the output of the GAP layer was directly fed into the output layer.

In the following year, Selvaraju et al. (2016) proposed a Gradient-based Class Activation Mapping (Grad-CAM) method to visualise classes in an image. This approach is a generalisation of CAM and could be applied to a variety of CNN networks which also includes fully connected layers. Here, the gradient of the final output class score, with respect to the feature map activations of the last convolutional layer, was computed and the global average pooled to get the important weights for regions of interest. We can then produce a visual heatmap overlaid on the actual images indicating the region of interest to the model (Figure 10).

We have used this Grad-CAM based approach to localise and highlight anomalies in our network because we had a fully-connected layer in our network. The heatmap produced in this approach indicates an anomalous region, if the classification output is flagged as anomalous. An example of a Grad-CAM output is shown in Figure 11, where the issue with soil moisture is identified by the coloured heat map lines overlapping the plot lines.

3.3 Results and discussion

We evaluated the results of this model using a blind validation set not used as part of the training or testing datasets. The validation set consisted of 477 weekly plots, generated over a period of three weeks from 159 different monitoring stations. We manually identified all the potential anomalies in the validation set and compared them against the results from the model. We used >=0.5

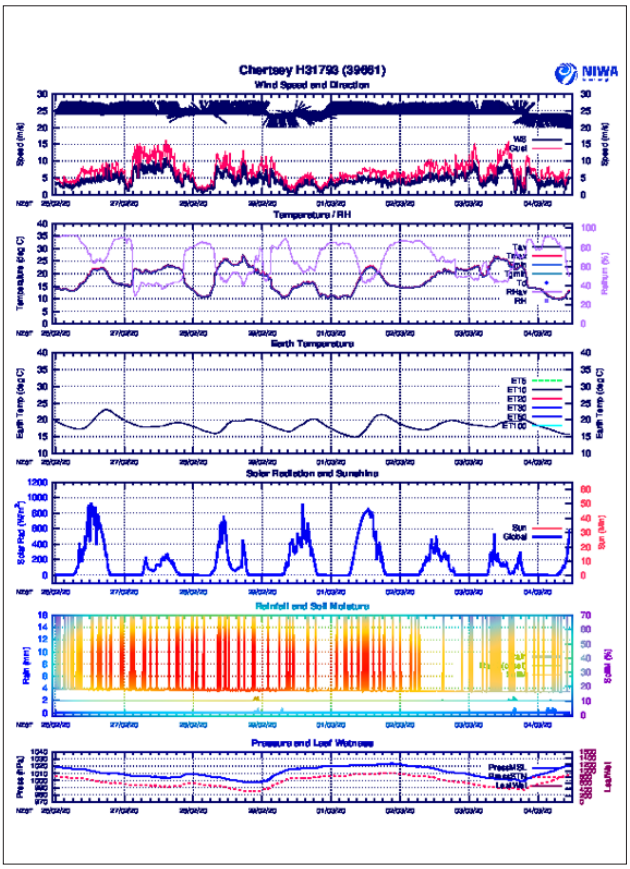


Figure 11: An example of a Grad-CAM output - a heat-mapped image. The anomalous region is highlighted in colour.

as the threshold probability score to classify an image as anomalous.

This threshold value was derived by plotting a Precision-Recall curve on the validation dataset for a range of thresholds (Figure 12). The plot showed that threshold values between 0.39 and 0.6 had high recall and good precision values. We chose the mid-point value of 0.5 as our threshold as that point had a higher recall value that did not significantly compromise the precision. As can be seen in the plot, the threshold could be lowered to around 0.4 to increase the recall at the expense of the precision. For operationalisation, we used a threshold of 0.5.

The Receiver Operating Characteristic (ROC) curve was plotted on the validation set and we calculated the area under the curve (Figure 13). The area was 0.98. This indicated that the model performance on the validation set is highly accurate.

In addition, we have used four scalar metrics, including F1 score and Matthews Correlation Coefficient (MCC) for scoring and evaluating the overall results (Table 1).

A perfect classifier would return an MCC score of +1; a classifier that always misclassifies would return a score of -1. In the table, all weeks have a high recall score, indicating that false negative rates were very low. Our overall precision was 0.88 indicating good positive predictive value. Precision was relatively low when compared to recall. Since we were building an anomaly detection system to identify issues with the data, our tolerance for false negatives was lower than for false positives. However, we are unable to keep the false positive tolerance too low because that would again involve considerable manual effort to go through the plots to reject false positives especially when we increase the scope of this process to cover additional stations and different time frequency data such as 10 minute and daily. Since the goal of the automated process was to save time and improve the efficiency of a QC review process, we have currently set a 20-25% tolerance on false positives and a 5-10% tolerance on false negatives. As seen in Figure 12, the threshold of 0.5 yields a precision of 0.88 and recall of 0.95. In future, we plan to test against new increasingly diverse validation datasets. During a test, if the precision of 0.75-0.8 increases the recall value significantly, we would choose an appropriate corresponding threshold. For operational runs, we aim to retain a recall score of at least 90% and a precision of at least 75%. As shown in Table 1, high values of F1 and MCC scores indicated a strong correlation between predicted and true classes.

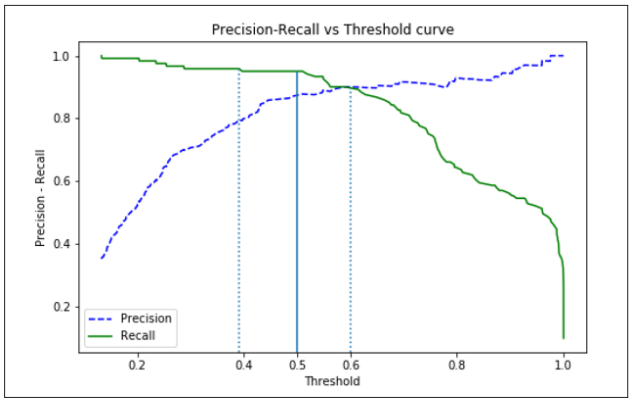


Figure 12: The precision and recall values plotted with respect to a range of thresholds. Dotted blue lines indicate the optimal range that could be used for thresholding that could improve either precision or recall accordingly. Threshold of 0.5 was chosen based on its optimal position with respect to good precision and a high recall value.

In addition to high classification quality, we also achieved high anomaly localisation accuracy using Grad-CAM. In

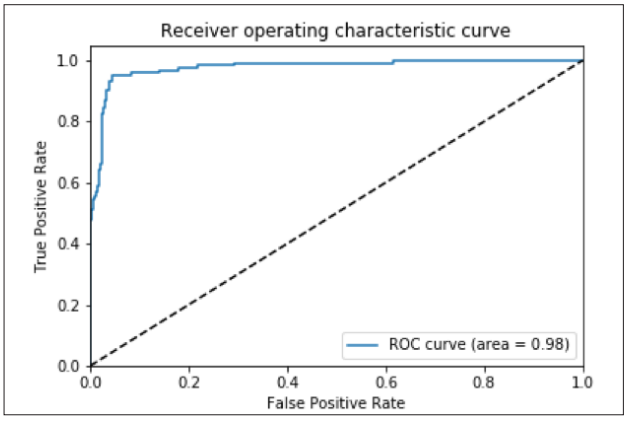


Figure 13: ROC curve plotted on the validation set

our validation set, the model identified 116 anomalous plots and all anomalous regions were successfully identified in 106 of these plots. So, we achieved 91% accuracy for anomaly localisation. In the remaining 10 images, anomalous regions were partially identified in nine images and, in one image, an anomalous region was incorrectly identified. In addition, anomalous regions were successfully identified for all the error types defined in Section 2. Examples of anomaly localisation can be seen in Figure 14.

Also, with the grad-CAM outputs of the validation set of all 477 images, we could see that the algorithm learnt with the plot labels present and that these were not detrimental to the performance of the algorithm.

As mentioned in section 2.3, Y axis scales were not constant in our dataset and the algorithm was able to classify and localise anomalies across all stations with varying scales.

These results suggest that the VGG-16 inspired anomaly classification model could be developed further and applied, on an operational basis, to minimise manual processing. We aim to further improve the model by detecting and adding more diverse scenarios to the training set and retrain the model. We could do this by regularly testing the output of the model, identifying scenarios where the model did not outperform the manual analysis, and then adding them to the training dataset.

During the manual review process, the QC issues that are identified or missed depends on the amount of the time spent reviewing each of these plots, along with the expertise of the manual reviewer. Currently there is no tracking of the QC issues missed during manual review and so the manual review process could not be scored directly. However, the labels of the validation set used in this study, to compare the ML process against, was manually labelled by an expert reviewer. So, Table 1 is an indirect, but fair comparison of a manual review process with this ML algorithm, with one caveat that a reviewer might not be in a position to spend sufficient time identifying and labelling these anomalies every week, as was done during this study. This manual labelling of the validation dataset by an expert reviewer was similarly done in Hundman et al. (2018), where the study used expert labelled dataset to test their LSTM based anomaly detection process.

Table 1: Statistics derived from the validation set, separated into individual weeks, and combined overall scores

	Precision	Recall	F1-score	MCC
Week 1	0.94	1.0	0.97	0.96
Week 2	0.87	0.91	0.89	0.85
Week 3	0.88	0.96	0.91	0.88
Overall	0.88	0.95	0.92	0.89

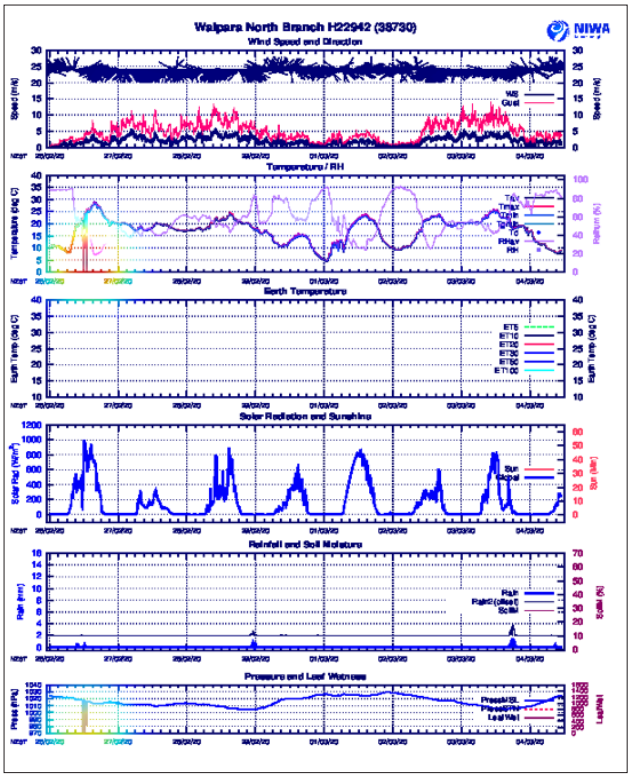


Figure 14: Examples of the different anomaly types defined in section 2.2. This image: example of sudden peak anomaly type correctly classified and highlighted by the Grad-CAM heatmap process.

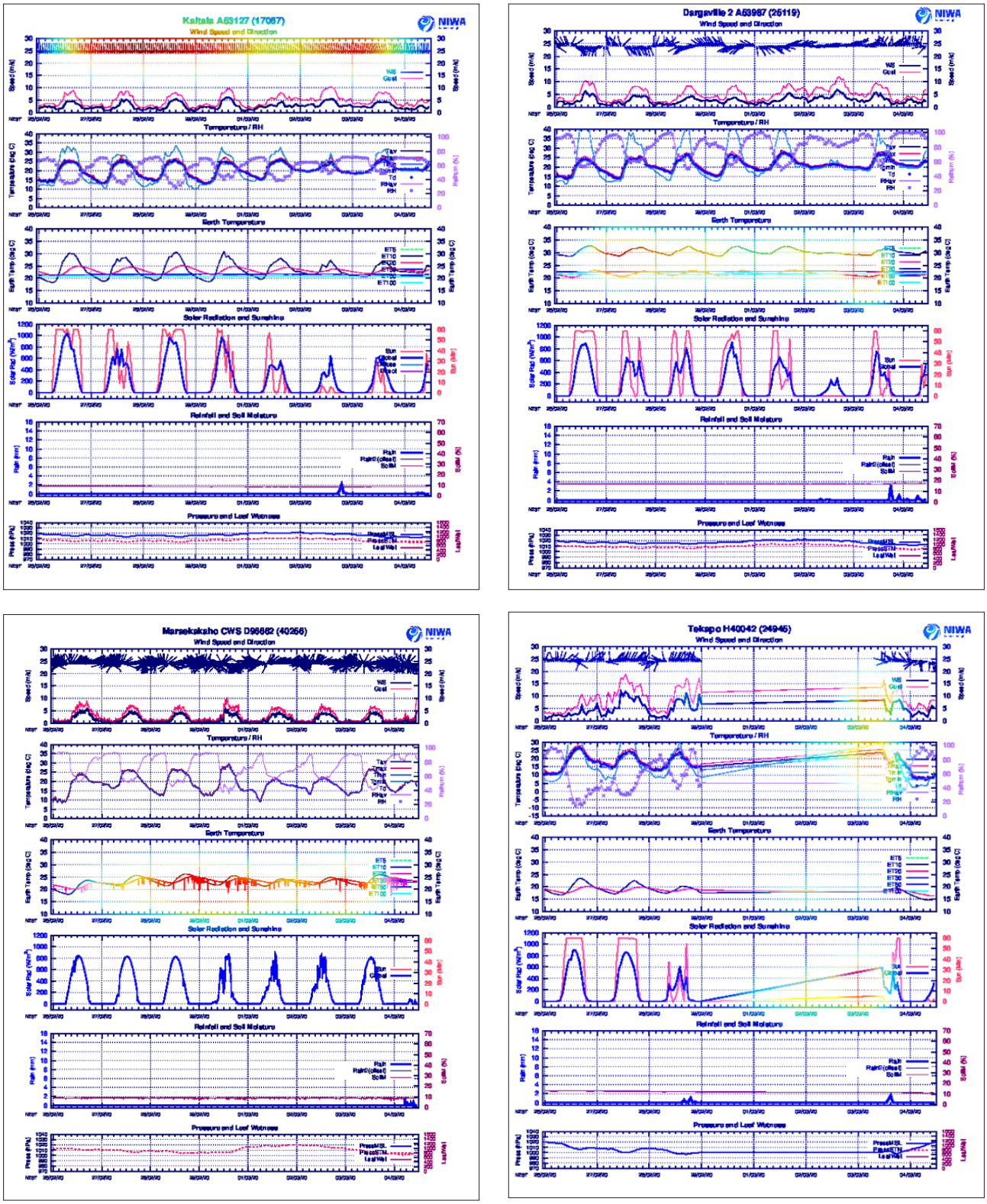


Figure 14 (continued): Examples of the different anomaly types defined in Section 2.2. From top left, examples of stuck instrument, uncorrelated behaviour, unusual blips and, bottom, incomplete timeseries. All these error types were correctly classified and highlighted by the Grad-CAM heatmap process.

4. Summary and Conclusion

In this study, we used a CNN image-based model to classify anomalies in the images of quality plots. To the best of our knowledge, this is the first attempt to use CNN to detect anomalous patterns in charts for QC purposes. From the archive of quality plots, we created a class-balanced dataset of around 1000 images for training our CNN model. This study investigated with a model using VGG-16 based architecture and were able to successfully train it using transfer learning. We were able to achieve high classification accuracy on our validation dataset, with an overall recall score of 0.95 and F1 score of 0.92. In addition, we were able to use a Grad-CAM based approach to successfully identify anomalous regions within images with an accuracy of 91%.

The above results indicate that this approach could be used on an operational basis to detect and identify anomalous plots and the specific anomalous regions within it. This method will reduce the significant effort of manually reviewing all the plots and thereby save significant time for the QC analyst. In addition, this will help to identify anomalies that might otherwise be overlooked or missed. This will improve the efficiency of the overall data quality process, as this enables the QC analyst to focus on data remediation instead of the identification of anomalies. Identification of data anomalies using this process will contribute to the improvement of the overall quality of the national climate data collection. This will in turn increase the reliability of climate data products and reports that are produced from the data extracted from the climate database. The procedure will also act as an early warning process to identify instrument issues and thereby enable more efficient planning of site visits.

This algorithm is currently operating within NIWA's CLiDB system as an 'assistant' for a weekly review of QC plots mainly on hourly and 10 min data for selected stations. This algorithm could be scaled up to detect anomalies at a greater number of sites. Also, this could be trained on different frequencies of timeseries like 1 min, 10 min, daily, sub-hourly. The algorithm could be scaled up to predict anomalies on different frequencies of a timeseries. The possibility of using this algorithm on a near real-time basis needs to be explored as this involves image generation and prediction. This algorithm falls short in generating an expected value for a variable in case of a missing observation or anomaly. As a next step, variable-specific individual timeseries algorithms can be explored that could complement this process by generating an expected value in the case of an anomalous observation. We also plan to compare the results of this study against standard variable specific timeseries-based algorithms to evaluate the relative performance of this CNN-based approach.

The work completed under this study so far has enabled us to operationalise this process in the climate database QC system. The introduction of this process has already considerably improved the efficiency of the anomaly detection process in the climate database system. As a next step, we intend to identify more scenarios of anomalies, such as those that this algorithm did not capture, and add them periodically to the training dataset for further model training. This feedback loop of identifying gaps and retraining will improve the overall accuracy of the model and the anomaly detection process. This retraining process will ensure the model can continue learning to identify increasingly diverse scenarios of anomalies. In addition, we are planning to expand this into a multi-class output model which would enable us to classify different error types for different variables. This would enable us to automatically produce anomaly reports based on different error types and parameters.

Acknowledgements

The authors would like to thank Andrew Harper, Alan Porteous and Errol Lewthwaite for their support of this work. We would also like to thank Dr. Kameron Christopher for his valuable inputs throughout this study.

References

Agarap, A.F. (2018). Deep Learning using Rectified Linear Units (ReLU). *ArXiv*, abs/1803.08375.

Balaji, A., Ramanathan, T., & Sonathi, V. (2018). Chart-Text: A Fully Automated Chart Image Descriptor. *ArXiv*, abs/1812.10636.

Bottou, L., Curtis, F.E., & Nocedal, J. (2016). Optimization Methods for Large-Scale Machine Learning. *SIAM Review*, 60, 223-311.

Chollet, F., & others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

Cliche, M., Rosenberg, D.S., Madeka, D., & Yee, C. (2017). Scatteract: Automated Extraction of Data from Scatter Plots. *ArXiv,* abs/1704.06687.

Cui, Z., Chen, W., & Chen, Y. (2016). Multi-Scale Convolutional Neural Networks for Time Series Classification. ArXiv, abs/1603.06995.

Davila, K., Setlur, S., Doermann, D., Bhargava, U. K., & Govindaraju, V. (2020). Chart Mining: A Survey of Methods for Automated Chart Analysis. I*EEE transactions on pattern analysis and machine intelligence*, 10.1109/TPAMI.2020.2992028. Advance online publication.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Li, F. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248-255.

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., & Söderström, T. (2018). Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. P*roceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.*

Hüsken, M., & Stagge, P. (2003). Recurrent neural networks for time series classification. *Neurocomputing*, 50, 223-235.

Inada, M., & T. Terano (2005). QC chart mining: extracting systematic error patterns from quality control charts. 2005 IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, HI, 2005, pp. 3781-3787 Vol. 4,

Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2018). LSTM Fully Convolutional Networks for Time Series Classification. *IEEE Access*, 6, 1662-1669.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *NIPS.*

Lin, M., Chen, Q., & Yan, S. (2014). Network In Network. *CoRR*, abs/1312.4400.

Liu, X., Klabjan, D., & Bless, P.N. (2019). Data Extraction from Charts via Single Deep Neural Network. *ArXiv*, abs/1906.11906.

Park, J. (2018). RNN based Time-series Anomaly Detector Model Implemented in Pytorch. Published in website URL: https://github.com/chickenbestlover/RNN-Time-series-Anomaly-Detection.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *ArXiv*, abs/1505.04597.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., & Li, F. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115, 211-252.

Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., & Batra, D. (2016). Grad-CAM: Why did you say that? Visual Explanations from Deep Networks via Gradient-based *Localization*.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556.

Wen, T., & Keyes, R.W. (2019). Time Series Anomaly Detection Using Convolutional Neural Networks and Transfer Learning. *ArXiv*, abs/1905.13628.

West. J., Venture. D., and Warnick. S., (2007). Spring research presentation: A theoretical foundation for inductive transfer. Brigham Young University, College of Physical and Mathematical Sciences.

Yang, J., Nguyen, M.N., San, P.P., Li, X., & Krishnaswamy, S. (2015). Deep Convolutional Neural Networks on Multichannel Time Series for Human Activity Recognition. *IJCAI*.

Zhao, B., Lu, H., Chen, S., Liu, J., & Wu, D. (2017). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28, 162-169.

Zheng, Y., Liu, Q., Chen, E., Ge, Y., & Zhao, J.L. (2014). Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks. *WAIM*.

Zhou, B., Khosla, A., Lapedriza, À., Oliva, A., & Torralba, A. (2016). Learning Deep Features for Discriminative Localization. 2*016 IEEE Conference on Computer Vision and Pattern Recognition* (CVPR), 2921-2929.