A Protocol for Creating and Modifying the Outlier Detection System's General Framework

D Divya (divya.d@cusat.ac.in)
Cochin University of Science and Technology
M. Bhasi
Cochin University of Science and Technology
M. B. Santoshkumar
Cochin University of Science and Technology

Research Article

Keywords: outlier, labelled data, neural network, hyper parameter optimisation, case study, nowcasting

Posted Date: May 31st, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1653583/v2

Abstract

Early detection of outliers has major applications in various industrial domains such as machine failure detection, patient monitoring systems and early warning systems of natural disasters. Although many machine learning and deep learning techniques have developed in the last few years, training of these algorithms requires a large amount of labelled dataset. However, generation of hand labelled training dataset is time consuming and has less accuracy. Likewise, the artificial dataset should follow the characteristics of real time data. Hence, generation of a time series dataset following the rules of data generation aids data scientists to develop and evaluate different supervised models. This generated data can help experts to develop a general framework for outlier detection. Hyper parameter optimisation performed in this base model ensures maximum efficiency of the generated model. Once the base model is ready, experts from various domains can utilise the framework for outlier detection in their specific application by applying domain specific heuristics. A case analysis using atmospheric dataset is conducted in this paper that act as a guideline for adding domain specific heuristics into the base model.

1. Introduction

Real time dataset contains various kinds of outlier instances with different statistical characteristics. Forecasting or early detection of these outliers require machine learning or deep learning models with high accuracy. Studies show that outlier detection can be considered as a special case of classification with two class labels: normal and outlier. Therefore, creating a labelled dataset will increase outlier detection accuracy (Bach et al., 2017). However, since outlier detection problems lack the presence of ground truth (Agarwal et al., 2017), the process of labelling requires either human expertise or programmatic approaches. All the studies related to generation of synthetic datasets in the outlier detection domain produces regular instances and outlier points (Steinbuss and Böhm, 2021; Emmott et al., 2015) without labels. Yet, labelled data generation is necessary to improve the key performance of machine learning or neural network algorithms (Bach et al., 2017).

This leads to a situation where there is a need for algorithms that can generate labelled artificial dataset. As a result, people search for alternative measures such as programmatic or other more efficient but noisier methods of producing training labels, sometimes referred to as weak supervision (Bach et al., 2017). Many of these rely on a single label source, or a small number of carefully selected and manually integrated sources (Mintz et al., 2009). However, labelling processes must ensure that outliers embedded in the normal data must follow certain rules set by earlier researchers in this field (Steinbuss and Böhm, 2021; Emmott et al., 2015). Some researchers developed techniques to assure that generated contains anomalies at different difficulty levels (Emmott et al., 2015) but none of these methods (Steinbuss and Böhm, 2021; Emmott et al., 2015; Chen et al., 2020) used the strategies to label the data that can be used for outlier classification. Although Ahamad et al. (2017) developed a benchmark named Numenta Anomaly Benchmark (NAB) dataset with labelled outliers, this dataset does not follow the real time data characteristics as proposed by earlier researchers. This necessitates the need for a data generation technique that has real time data characteristics.

As most of the real time applications exhibits characteristics of a time series dataset, data generation model with similar features has to be selected for this analysis. Random data distribution is one of the simplest and most significant models in time series forecasting (Nau, 2014) which can represent data with high variability (Nau, 2014). In order to add outliers into the data earlier researchers used uniform distribution with various difficulty levels and frequencies (Steinbuss and Böhm, 2021; Emmott et al., 2015; Chen et al., 2020).

Once the data is generated successfully, machine learning or deep learning can be used to build models to detect outliers in the data. Accepting the fact that neural networks have higher detection rates as compared to the machine learning counterparts, we developed a neural network model to perform outlier detection. Hyper parameter optimisations are performed in the base model to attain maximum efficiency. This base neural network act as a general framework which can be used in different industrial domains with domain specific adaptations. Thus, this research work is designed to address the following research questions.

R1

Is it possible to generate a labelled dataset that has the same characteristics as that of real time dataset?

R2

How to develop a base neural network model which can be used for outlier detection in multiple domains?

R3

What is the procedure used to modify the base model with domain adaptation?

These research questions are addressed using a three-stage process. First stage generates a dataset similar to the real time data instances. Second stage base neural network model that can be used across multiple domains. This base model is optimised with the help of a generated dataset which will save the time of hyper parameter tuning. In the third stage domain expert knowledge is used to modify and enhance the model. Case study explained in Section 5 helps the user to understand how to incorporate domain specific features into the system.

Thus, there are three advantages for this research work. Firstly, this research work will help researchers across various industrial domains to utilise the first step for generating a labelled dataset. Secondly, it creates a general framework for outlier detection without further processing. This helps users to save time involved in hyper parameter tuning of the neural network. Moreover, this paper acts as a guideline for users from various industrial domains on how to add domain specific knowledge to modify and enhance their model.

Rest of the paper is organised as follows. Section 2 reviews the literature that helped us to build the base model. Section 3 details the research methodology and Section 4 presents data generation results; Section 5 illustrates methods to apply the base neural network model in a real time application through a case study.

2. Literature Review

2.1 Need for synthetic data generation

Outlier detection has many applications in various domains such as process monitoring (Wang and Mao, 2019; Xu and Ding, 2022), intrusion detection (Huang and Lei, 2020), heart failure detection (Hammad et al., 2021), environmental hazard monitoring (Prodhan et al., 2022) and water intake monitoring (Xue et al., 2019). In all these studies the notion of 'outlier' varies from context to context (Yepmo et al., 2022). For example, irregular heart rhythm represents outliers in heart failure detection whereas natural hazards like flood or drought represents outlier instances in environmental monitoring. Since the data associated with these applications is organised in a time series pattern, approaches that can detect outliers in time series data are chosen. Survey done by Gupta et al. in 2014 help users to classify outlier detection methods used for time series dataset (Gupta et al., 2014).

In the literature, many of these time series applications use unsupervised methods such as clustering (Wu et al., 2016), distribution base methods (Yamanishi et al., 2004), distance-based methods (Ramaswamy et al., 2000) for detecting outliers. Zimek et al. 2012 reviewed unsupervised outlier detection techniques for high dimensional numerical data (Zimek & Kriegel, 2012). However, all these algorithms face a key problem: real-time users of these algorithms require explanations for their outputs, a problem known as eXplainable Artificial Intelligence (XAI) (Yepmo et al., 2022). But due to the unavailability of ground truth for outlier detection problems (Agarwal et al., 2017) it is difficult to explain the reason for being an outlier. Bach et al., 2017 uses weak supervision to label the outliers which helps to differentiate outliers and normal data objects that explain outlier points on the basis of the domain it appears. However, according to Yepmo et al., 2022 meaning of outliers varies at different domains. Therefore, synthetic data generation is essential to get labelled data that can be used across multiple industrial domains. Next section details algorithms developed for synthetic data generation in the area of outlier detection.

2.2 Synthetic data generation in outlier detection problems

Numerous studies are available in synthetic data generation that have outlier points embedded into it. A few of these studies are restricted to domain specific data generation such as data related to wireless sensor networks (Fan et al., 2004), financial dataset (Gonzalez et al., 2002), intrusion detection (Pham et al, 2014) and detection of outliers in image dataset (Steinwart et al., 2005).

As discussed, due to the fact, the meaning of outliers varies in different industrial domains, a general framework for artificial data generation is essential. Steinbuss and Bohm, 2021 devised a general framework for creating artificial outliers based on a set of rules imposed by previous researchers in the

field (Emmott et al., 2015; Chen et al., 2020). His research used a technique called 'sampling from a distribution' to generate regular examples and then inject outliers using uniform distribution at various difficulty levels. Similar to this, Emmott et al., 2015 used multivariate Gaussian distribution for regular dataset generation and Domingues et al. (2018) generated data from two separate T distributions. However, it is important to produce data points that has the similar characteristics as that of real time data points. As many of the major applications areas of outlier detection contain time series dataset; generated data instances also must follow the characteristics of a time series dataset. Also, synthetic data generation process should ensure that algorithms developed to detect outliers performs well with diverse set of data points (Sánchez-Monedero et al., 2013). But all the existing algorithms generate time series dataset (Zulkipli et al., 2021) without labels. The establishment of the Numenta Anomaly Benchmark (NAB) (Ahmad et al., 2017), a labelled benchmark, is another noteworthy contribution in this area. However, NAB and its related works do not cover important characteristics such type and frequency of outliers in the data. Hence, there is need to address the first research question, R1: Is it possible to generate a labelled dataset that has the same characteristics as that of real time dataset?

2.3 Hyper parameter optimisation in neural network

Data generation is followed by selection of an appropriate outlier detection technique. Because of the greater outlier identification rate, neural networks and deep learning algorithms are used in the majority of outlier detection techniques (Zeng et al, 2020). Still, these networks require hyper parameter optimisation techniques for enhancing the detection rate. Existing research in this area has primarily focused on improving the outlier detection rate for a given application (Pfülb, 2019) or a specific network segment (Dhaouadi et al., 2019). Furthermore, there is a scarcity of domain adaptation approaches (Najafabadi et al., 2015), where learned a model works well in the targeted domain (Zhao et al., 2022). However, it is important to develop a neural network architecture with optimised hyper parameters that can be used in a variety of applications. This forms the basis for the research question R2.

2.4 Early warning systems

Once the base model is developed, this model can be enhanced to use in a domain specific application. Current research works focuses on nowcasting approaches that predict outliers that will occur during the next few hours (Brandyn et al., 2018). This nowcasting approach has a significant impact in real-time scenarios such as dam deformation early warning (Chen et al., 2021), and gearbox failure warning (Wang et al., 2021). Domain specific features largely influence accuracy of these systems (Diao et al., 2020). Therefore, adding domain specific components while developing a time series forecasting is a good approach to increase the accuracy of these systems (Chen et al., 2021). As outlier detection causes a problem of class data level strategies such as oversampling and under sampling of data instances can also help the users to achieve higher accuracies (Johnson and Khoshgoftaar, 2019).

3. Methodology

3.1 System Architecture

In Fig.1 the data generation engine takes three parameters. First, the statistical distribution of regular instances then the distance and frequency metrics of outliers. Literature resources reveal that these three parameters form the major components for data generation with outliers. Unlike the data generation process used in the literature the proposed architecture generates a labelled dataset. Once the data is generated successfully, this data is stored inside a database that can be accessed by any machine learning or deep learning model for analysis. In the current analysis, after generation of the dataset a neural network model is created that forms the basic model for outlier classification. Hyper parameter optimisation is done in this base model so that the optimised model can act as a general framework for further analysis. Once the optimised model is ready to use, domain specific users can make use of the general framework for outlier classification in their domain specific dataset. Second stage of the proposed method uses domain specific heuristics to modify the general framework or the base model to improve the model for their specific problem. This paper illustrates the design of the base model for outlier classification and the method used to modify the base model for a specific application using domain specific heuristics.

3.2 Components of data generation algorithm

This study compares four different types of data instances. Two types of genuine data points which can be collected from any real time dataset. Secondly, two artificial data points generated by the data generator. Based on the genuine data points artificial data instances are generated for the current analysis (Steinbuss and Böhm, 2021).

Studies revealed that most of the real time systems such as process monitoring (Wang and Mao, 2019), intrusion detection (Huang and Lei, 2020), heart failure detection (Hammad et al., 2021) and water intake monitoring (Xue et al., 2019) contains time series dataset. During the process of data generation, the generated data should follow the same statistical characteristics as that of time series data. Since a random distribution can better represent a time series dataset, random distribution can be used for generating regular instances. A separate uniform distribution is used to insert outlier points into the dataset with various difficulty levels having different frequencies. Notations used in the proposed algorithm is given in Table 1.

Table 1: Components of the data generation algorithm

Notations used in the algorithm
Vmin: Minimum limit for the generated data
Vmax: Maximum limit for the generated data
current value: current value of outlier instance
ss: step size
run: current
max: maximum run size
min: minimum run size
direction: checks direction of next move
mistake: randomly generated outlier value

3.3 Data generation algorithm

RAND_GEN (Min, Max)

Step 1: Generate Seed (VMin,Vmax) as current value

Step 2: Set maximum run size as 'max' and step size as 'ss' for this run

Step 3: if current value > max:

Step 3.1 output = bringdown.bringdown(currentvalue,ss,run)

Step 3.2 current value = output

Step 3.3 outputlist = [j, output, 0]

Step 3.4 row = [j, output]

Step 3.5 Output the current value to csv file

Step 4: If the current value < min:

Step 4.1 output=bringup.bringup(current value,ss,run)

Step 4.2 current value = output

Step 4.3 outputlist = [j, output, 0]

Step 4.4 row = [j, output]

Step 4.5 Output the current value to csv file

Step 5: Generate random number r using random.random()

Step5.1 if (r>0.5):

Set the direction as 1

Step 5.2 else:

Set the direction as 0

Step 5.3 if direction is 1:

- *output = current value + ss*
- *current value = output*
- *ss = stepsize.stepsize()*
- *row = [j, output]*

Step 5.4 Output the current value to csv file

Step 6: Call insert outliers using Call uniform dist (a, b)

Step 6.1 mistake = random.random()

*Step 6.2 Gen-Localanomaly(2*ss*2*run), (2*ss*2*run);*

- *a = current value + (ss *2*run *2*mistake)*
- *b = current value - (ss *2*run * 2*mistake)*
- *rv = round(random.uniform(a,b))*
- *return rv*

*Step 6.3 Gen-globalanomaly((3*ss*3*run), (3*ss*3*run)).*

- *a = current value + (ss *3*run *3*mistake)*
- *b = current value - (ss *3*run * 3*mistake)*
- *rv = round(random.uniform(a,b))*
- *return rv*

Once the data is generated the next step is to verify whether the data generation process is correct or not. This is done using Chi Square analysis in Section 4.2

4. Results

4.1 Data generation Results

As discussed in Section 3, a dataset which is similar to the time series data is generated. User interface is designed to take input from the user to set the size and limit for the generated data. Here, data size determines the number of data points generated and limit value determines minimum and maximum limits for the generation. Fig.3 gives the user interface. Based on user input, the simulator will generate a dataset using the algorithm given in Section 3.2.

The generated data will be stored inside a database. Users can retrieve this generated data for further analysis. Fig.4 depicts data set which is retrieved from the database

Fig. 5 plots histogram of the generated dataset which is retrieved from the database.

4.2 Verification of the generated dataset

As discussed in Section 3.1, based on literature (Steinbuss and Böhm, 2021) for the current analysis outliers are inserted based on two parameters: distance and frequency. Distance metric decides whether the outliers are local or global. Distance metric rules are verified using distance measures given by pioneers in the field (Steinbuss and Böhm, 2021). In the current analysis, local outliers are generated at 2*stepsize*2*run and global outliers are generated at a distance of 3*stepsize*3*run.

In order to check whether the generated follows random data distribution, Augmented Dickey Fuller (ADF) test is used to check whether the generated dataset follows a random distribution

4.3 Creating a base neural network model to classify outliers in the generated data

Once the data is generated, the generated data can be used to implement a base neural network model to perform outlier classification. Hyper parameter optimisation of the model is done using the generated data. This forms the base model for further analysis, where users from various industrial domains can utilise the optimised base model for their specific domain.

4.3.1 Structure of base neural network model

Fig. 6 gives structure of the base neural network developed. This base model act as the building block for further domain specific improvements.

4.3.2 Hyper-parameter optimisation

Objective of hyper parameter optimisation is to identify best values for number of epochs and batch size. We adopted trial and error strategy for hyper parameter optimisation.

Table 2: Hyper parameter optimisation

Trial Number	Network Structure	Results
1	Epoch 10, Batch Size 16 (10 executions)	**Outlier detection Rate:** Mean:52.49 Std. Dev.: 31.49 **False alarm rate:** Mean: 8.86 Std. Dev: 8.81
2	Epoch 50, Batch Size 16 (10 executions)	**Outlier detection Rate:** Mean:62.53 Std. Dev.: 19.4 **False alarm rate:** Mean: 8.8 Std.Dev:8.9
3	Epoch 100, Batch Size 16 (10 executions)	**Outlier detection Rate:** Mean:79.22 Std. Dev.: 10.1 **False alarm rate:** Mean: 13.5 Std.Dev:6.0
4	Epoch 100, Batch Size 40 (10 executions)	**Outlier detection Rate:** Mean:76.79 Std. Dev.: 16.1 **False alarm rate:** Mean: 12.4 Std.Dev:7.76

From the given table, 100 epochs and a batch of size 16 has a high detection rate and low false alarm rate. Therefore, trail number 3 can be selected as the optimised base neural network model. In order to ensure that users can attain a benchmarking efficiency from the base model, a t_test (Kim et al., 2015) is conducted to confirm that using selected neural network model outlier detection rate of 75 % is attainable for any dataset.

H_0: Outlier detection rate of any real time dataset with the selected neural network model is greater than or equal to 75

H_1: Outlier detection rate of any real time dataset with the selected neural network model is less than 75

To verify the correctness of this assumption a single sample t_test can be used.

For 100 epochs and a batch of size 16, sample mean value of 10 executions is 79.22 with standard deviation of 10.1. Hypothesis claims that using this optimised set of parameters users can attain at least 75% outlier detection rate.

$$\text{t\_test statistic} = \frac{\bar{x}-\mu}{S/\sqrt{n}}$$

Table 3: t_test Results

$\bar{x}-\mu$	$S/\sqrt{n}$	t_test statistic
4.22	10.11/3.16=3.196	**1.320**
α	0.01	
Degrees of freedom	10-1 =9	

Here, t_test statistic is 1.320<1.833(critical value)

Hence, t_test proves that null hypothesis can be accepted. This verifies that any neural network model with the selected hyper parameters will have a performance greater than or equal to 75.

5. Case Study

5.1 Using neural network model for weather prediction in ACARR dataset

A case study is undertaken utilising an atmospheric dataset to validate the hypothesis presented in Section 4.3. Dataset for the case analysis is collected from Advanced Centre for Atmospheric Radar Research (ACARR) located at Cochin University of Science and Technology (CUSAT), India. Advanced Centre for Atmospheric Radar Research (ACARR), Cochin University of Science and Technology (CUSAT) is the most modern and advanced ST wind profiler radar to better comprehend diverse troposphere circulation aspects and lower stratosphere, as well as their impact on the underlying environment the Indian subcontinent's monsoon circulation. For the current analysis, event rain is considered as outlier points. For the current case analysis base neural network obtained from simulation is used to classify the event rain.

After applying the base model for outlier detection domain specific adaptations can be done in this base model. Current section presents a case study using the basic features. For getting the basic features in the domain a literature study is conducted. Based on that, rainfall rate is dependent on the thermodynamic variables (Zawadzki et al. (1981). Demographic studies show that the rainfall over Cochin is influenced by both the coastal effect and the orographic effect due to its proximity to both the Sea and the hills. ACARR has most modern and advanced ST wind profiler radar to better comprehend diverse troposphere circulation aspects and lower stratosphere, as well as their impact on the underlying environment of the Indian subcontinent's monsoon circulation. The cloud database of the centre stores all the atmospheric features. Yet, studies show that temperature, humidity, wind speed, pressure and radiation are the relevant factors (Jaseena et. al., 2020; Mathew et al., 2021; Mohankumar et al., 2019) that affect the event rainfall. Because of that, these relevant features are taken for the current analysis. As discussed, the outlier represents the event ‘Rain’ and normal points correspond to ‘No Rain’ events. Atmospheric dataset with 17520 data instances for the year 2018 is used for this analysis. Base neural network with hyper parameters obtained from Table 2; 100 epochs and 16 batch size are utilised for this case study.

Rain data classification is done using the set of features given in Table 4.

Table 4: Basic set of features used for rain data classification

Sl. No.	Basic features
1	Temperature
2	Humidity
3	Wind speed
4	Pressure
5	Radiation

Table 5: Results: Weather Prediction

Data and Results: Weather Prediction
No of normal points (16133, 7)
No of outliers (1387, 7)
No of outliers correctly classified 220
True Positive 2299
False Positive 920
False Negative 61
Accuracy of the system: 71.97142857142858
Outlier Detection rate 78.29181494661922
False Alarm rate 28.580304442373405

In Table 5 outlier detection rate corresponds to rain data classification. Results shows that outlier detection rate is in accordance with the hypothesis developed in Section 4. This validates the use of the neural network with a set of hyper parameters obtained from the generated dataset. Once the general framework for outlier classification is created, the same neural network model can be extended further to perform domain specific analysis.

5.2 Extending the base neural network model to nowcasting model

5.2.1 Nowcasting Model Development

The case study explained in Section 5.1 can be considered as a weather forecasting system with zero lead time (current weather prediction). Current weather prediction uses current atmospheric features to predict the likelihood of event rain in the current time frame. However, nowcasting is an important term used in meteorology for forecasting the weather occurring in the next few hours (Brandyn et al., 2018). Base model obtained and utilised in section 5.1 can be extended to a nowcasting model. This section presents a case study on how to use the neural network model for weather nowcasting after adding new derived features into the base weather prediction model. The objective of this case study is to extend the base model developed using the generated data by adding atmospheric features obtained from domain experts.

This case study uses an architecture of the IoT system illustrated in Fig.7. Here, data collected from the radar (sensor in other applications) in the perception layer is transmitted to the cloud through a network layer. Based on the user's requirement, selected data can be downloaded into a local database for processing and prediction purposes. Business layer visualises the results which help the users to interpret the results. This IoT architecture can be viewed as a general model for any application with domain specific changes in the dataset.

To develop a nowcasting system for this architecture literature has been studied. It shows that several data driven approaches have been developed in recent years to predict rainfall (Manandhar et al., 2019). However, domain experts explain that rainfall is dependent on a myriad of atmospheric parameters (Manandhar et al., 2019) and a single feature cannot increase forecasting accuracy. Liu et al., 2019 emphasises this fact that a good model cannot be constructed if the neural network model is dependent on a single predictor. Therefore, addition of domain specific features into the basic neural network model can enhance the rain fall forecasting accuracy. Since there is no strong correlation between rainfall and any of these meteorological parameters (Liu et al., 2019); new techniques or features must be identified to improve the forecast accuracy of short-term rainfall. Studies show that current rainfall information of different magnitudes has an impact on upcoming floods (Jia et al., 2020). In accordance with this study a new feature current rain is introduced for this analysis. Time of the day used by Liu et al., 2019 is modified into a categorical data due to its impact on the prediction accuracy. Similarly, seasonal factors are included in accordance with study conducted by Ceglar and Toreti, 2021 and the studies about Indian sub-continent Kothawale et al., 2010.

This forms the set of three derived features: current rain, four separate categories for time of day (12am-6am, 6am-12pm, 12pm-6pm, and 6pm-12am) and four different classes for season of the year (Winter: Jan-Feb, Pre-Monsoon- March-May, Summer Monsoon- June-Post Monsoon-Oct-Dec). Table 6 presents the domain specific features used for the current analysis.

Table 6: Selected features for nowcasting

Sl. No.	Basic features	Derived features
1	Temperature	Current rain
2	Humidity	Season of the Year (4 classes)
3	Wind speed	Time of the day (4 classes)
4	Pressure	
5	Radiation	

After selection of domain specific features, base neural network model can be used directly without any modifications.

5.2.2 Nowcasting: Results

In this analysis, the base neural network model developed in section 5.1 is modified to perform forecasting of rain events in different lead times. Because nowcasting is used for prediction for 0- 6hrs, analysis done in this case study examines the forecasting accuracies for 0-6hrs. Table 7 presents the nowcasting results after applying domain specific heuristics into the base model. The fundamental neural network model employed in this research is the same as in Section 5.1, and the dataset is the same: 17520 data instances for the year 2018.

Table 7: Forecasting accuracies with various lead times

Forecasting results			**Prediction Results from Table 7**
Features: Time, Season, Temperature, Humidity, Wind Speed, Wind direction, Pressure, Radiations, Current Rain			Temperature, Humidity, Wind Speed, Wind direction, Pressure, Radiations
Simple NN(T+N)	**Rainfall forecasting rate**	**Rain fall forecasting False alarm rate**	**Current:** Outlier detection rate:78.2 False alarm rate: 28.5
Current	82.4	24.2	
1hr	81.2	34.3	
2hrs	80.4	29.4	
3 hrs	80.6	29.5	
4 hrs	76.4	32.2	
5 hrs	78.2	31.9	
6hrs	74.5	24.0	

Results and comparisons presented in Table 7 shows that addition of derived features improved the forecasting accuracy of the base model. This case study illustrates the various steps to apply the base model for short term rainfall forecasting. Results show that within a lead time of 6 hrs existing neural network models have a higher forecasting rate with tolerable false alarm rates.

5.2.3 Data level improvements in nowcasting accuracies

For any data analysis problems improving forecasting accuracies is a necessary step that helps the users to enhance the basic model. As we need the same structure of the base neural network model with the same set of hyper parameters, data scientists have to ponder for new measures to increase the accuracy. Due to the fact that the dataset used in this analysis contains class imbalanced dataset, where the percentage of data instances in the class ‘rain’ is low as compared to the ‘no rain’ data instances; further optimisations can be done at data level. Data-level methods for addressing class imbalance problems include oversampling and under-sampling. (Johnson and Khoshgoftaar, 2019). Under-sampling discards data voluntarily, lowering the overall amount of data from which the model may learn. Due to the increasing size of the training set, over-sampling will result in an increase in training time (Chawla et al., 2004). This case analysis uses both over sampling and down sampling methods to improve the nowcasting accuracies. Table 8 and Table 9 and gives accuracy improvements after up sampling and down sampling for a lead time of 1 hr and 6 hrs.

Table 8: Comparison of nowcasting accuracies (Lead time: 1 hr)

Season	Percentage of rain data	Rain data Detection Rate	False Alarm Rate
Without sampling	8	81.2	34.3
Down sample	1:1	98.7	69.7
Up sample: Random	1:1	97.5	57.7

Table 9: Comparison of nowcasting accuracies (Lead time: 6 hrs)

Season	Percentage of rain data	Rain data Detection Rate	False Alarm Rate
Yearly	8	74.5	24.0
Down sample	1:1	97.2	76.8
Up sample	1:1	97.2	69.8

Oversampling and down sampling methods increase the detection rate penalising the false alarm rate, further analysis is required in this dataset. As a result, researchers are looking into the impact of a higher number of outlier points (event rain) on forecasting accuracy. For an atmospheric dataset, this can be accomplished by data analysis that divides the data into different seasons.

Table 10: Season Wise Analysis (Lead time: 1 hr**)**

Season	Percentage of rain data	Rain data Detection Rate	False Alarm Rate
Monsoon	16.9	91.0	43.7
Autumn	7	54.2	4.0
Summer	3.8	51.4	5.0
Winter	0.6	5.0	3.4

Table 11: Season Wise Analysis (Lead time: 6 hrs**)**

Season	Percentage of rain data	Rain data Detection Rate	False Alarm Rate
Monsoon	16.9	87.2	53.5
Autumn	7	54.7	23.2
Summer	3.8	22.2	6.7
Winter	0.6	0	0

Results obtained from season wise analysis shows that forecasting the event rain is easier in monsoon as compared to other seasons. This identification reveals the fact that identifying outliers is easier when the number of outlier instances are high. This discovery can be used by data analysts to improve forecasting accuracy in domain-specific datasets.

6. Discussion

This research work can be considered as a three-stage process to address three research questions as depicted in figure 7.

First two stages of this research work present a cross domain method that can be utilised by people from various industries. Third stage acts as a knowledge framework for domain specific users. Detailed discussion given in Section 6.1 gives theoretical implication of the study and section 6.2 explains the practical implications of this research.

6.1 Theoretical implications

Researchers in this domain have the following advantages.

- Generation of labelled outliers help data scientists to utilise this data for their specific application.
- Base neural network model developed with optimised hyper parameters helps users across the industrial domains to make use of the general framework for outlier detection without further analysis. Hence, data scientists can save the time for initial model development and optimisation process.
- Figure 6 details the layered architecture of an IoT system that aids the algorithm developers to identify their specific layer of research.
- Case studies presented in this work act as guidelines for researchers across industrial domains to identify relevant features in their specific domain.
- Case study guides computer scientists on how to modify the general base model for their specific application.
- Data level improvements in nowcasting accuracies done in the case analysis helps data scientists to use the same set of enhancements in their particular dataset without further analysis.

Thus, this study builds a general framework for data scientists to develop an early warning system in their specific application domain.

6.2 Practical implications

Another set of beneficiaries who take benefits of this research work includes algorithm users.

- Since the developed model presents a general framework, users from various industrial domains can use this base model as a benchmarking model for their outlier detection studies.

- With help of expert knowledge algorithm users can extract domain specific features and can enhance base model

7. Conclusion

Outlier detection is an important problem that helps users to identify abnormal data instances in the data. Identification of these abnormal instances may be a time-consuming task for data analysts. Although many unsupervised algorithms have been developed for outlier identification, supervised algorithms show better detection accuracies in majority of the cases. This leads to a situation where there is a requirement for a labelled dataset for training. Even with labelled data, the training and optimisation process also takes a huge amount of time. Seeing these, this research work presents a data generation algorithm which produces labelled data that has similar characteristics as that of a real time dataset. This generated dataset is used to develop and optimise the base neural network that can be used for outlier detection. The algorithm can be further extended to early warning systems after modification of the base model. A case study has been demonstrated using an atmospheric dataset that guides users to apply the same model in various industrial domains. This research work helps both researchers and practitioners to develop an effective outlier detection system in their specific domain.

There are some limitations for this research work. Firstly, the base model uses a neural network algorithm, other supervised models are not explored in this study. Further improvements in the base model requires domain expertise which may need further exploration in the specific domain.

Declarations

Ethical Approval and Consent to participate

Not Applicable

Consent for publication

I, Divya D, give my consent for the publication of the manuscript in 'Journal of Intelligent Information System'.

Availability of supporting data

Generated data is available on request. Advanced Centre for Atmospheric Radar Research (ACARR) data is not publicly available.

Competing interests

The authors have no relevant financial or non-financial interests to disclose.

Funding

There is no funding

Authors' contributions

Divya D: Data collection, implementation and validation, paper writing

Dr. M. Bhasi: Concept design, verification of results and guidance for paper writing

Dr. Santoshkumar M.B.: Guidance for implementation and paper writing

Acknowledgments

The authors express their deepest gratitude to Advanced Centre for Atmospheric Radar Research (ACARR) in Cochin University of Science and Technology, Kochi, Kerala, India, supported by Ministry of Earth Sciences (MoES), Govt. of India for providing data for this study.

References

1. A. Zimek, E. Schubert, H.-P. Kriegel, ‘A survey on unsupervised outlier detection in high-dimensional numerical data’, Statistical Analysis and Data Mining, 5 (5) (2012), pp. 363-387
2. Ahmad, S., Lavin, A., Purdy, S. and Agha, Z. (2017), ‘Unsupervised real-time anomaly detection for streaming data’, Neurocomputing, Vol. 262, pp.134–147, DOI: 10.1016/j. neucom.2017.04.070.
3. Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In Advances in neural information processing systems. 3567–3575.
4. Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. Proceedings of the VLDB Endowment 11, 3 (2017), 269–282
5. Bae WD, Kim S, Park CS, Alkobaisi S, Lee J, et al. (2021) Performance improvement of machine learning techniques predicting the association of exacerbation of peak expiratory flow ratio with short term exposure level to indoor air quality using adult asthmatics clustered data. PLOS ONE 16(1): e0244233. https://doi.org/10.1371/journal.pone.0244233
6. C. C. Aggarwal, "Outlier Ensembles", Outlier Analysis, 2017.
7. Ceglar, A., Toreti, A. Seasonal climate forecast can inform the European agricultural sector well in advance of harvesting. npj Clim Atmos Sci 4, 42 (2021). https://doi.org/10.1038/s41612-021-00198-3
8. Chawla NV, Japkowicz N, Kotcz A. Editorial: Special issue on learning from imbalanced data sets. SIGKDD Explor Newsl. 2004;6(1):1–6. https://doi.org/10.1145/1007730.1007733.
9. Chen, D., Lu, CT., Kou, Y. et al. On Detecting Spatial Outliers. Geoinformatica 12, 455–475 (2008). https://doi.org/10.1007/s10707-007-0038-8

10. Chen, Z., Luo, X. and Sun, Y. (2020) 'Synthetic data augmentation rules for maritime object detection', International Journal of Computational Science and Engineering, Vol. 23, No. 2, pp.169–176, DOI: 10.1504/ IJCSE.2020.110541.
11. D. Jeske, B. Samadi, P. Lin, L. Ye, S. Cox, R. Xiao, T. Younglove, M. Ly, D. Holt, and R. Rich. Generation of synthetic data sets for evaluating the accuracy of knowledge discovery systems. In Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pages 756–762. ACM, 2005
12. D. Jeske, P. Lin, C. Rendon, R. Xiao, and B. Samadi. Synthetic Data Generation Capabilties for Testing Data Mining Tools. MILCOM, 0:1–6, 2006.
13. Domingues, R., Filippone, M., Michiardi, P. and Zouaoui, J. (2018) 'A comparative evaluation of outlier detection algorithms', Pattern Recogn., February, Vol. 74, No. C, pp.406–421, https://doi.org/10.1016/j.patcog.2017.09.037
14. Emmott, A., Das, S., Dietterich, T.G., Fern, A. and Wong, W. (2015) A Meta-Analysis of the Anomaly Detection Problem, arXiv: Artificial Intelligence.
15. F Gonzalez, D Dasgupta, and R Kozma. 2002. Combining Negative Selection and Classification Techniques for Anomaly Detection. In Proceedings of the 2002 Congress on Evolutionary Computation. (CEC '02). IEEE, Los Alamitos, CA, 705–710. https://doi.org/10.1109/CEC.2002.1007012
16. Foyez Ahmed Prodhan, Jiahua Zhang, Shaikh Shamim Hasan, Til Prasad Pangali Sharma, Hasiba Pervin Mohana,A review of machine learning methods for drought hazard monitoring and forecasting: Current research trends, challenges, and future research directions, Environmental Modelling & Software, Vol. 149,2022,105327, doi: 10.1016/j.envsoft.2022.105327
17. G. Albuquerque, T. Lowe and M. Magnor, "Synthetic Generation of High-Dimensional Datasets," in IEEE Transactions on Visualization and Computer Graphics, vol. 17, no. 12, pp. 2317-2324, Dec. 2011, doi: 10.1109/TVCG.2011.237.
18. Georg Steinbuss and Klemens Böhm. 2021. Generating Artificial Outliers in the Absence of Genuine Ones — A Survey. ACM Trans. Knowl. Discov. Data 15, 2, Article 30 (April 2021), 37 pages. DOI: doi.org/10.1145/3447822
19. Ingo Steinwart, Don Hush, and Clint Scovel. 2005. A Classification Framework for Anomaly Detection. Journal of Machine Learning Research. 6 (Feb. 2005), 211–232
20. J. Dhaouadi, M. S. Aktas, O. Kalipsiz and E. Balcik, "On the Use of Hyperparameter Optimization in Big Data Processing Pipelines: A Case Study," *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 2019, pp. 1-5, doi: 10.1109/ASYU48272.2019.8946352.
21. Johnson, J.M., Khoshgoftaar, T.M. Survey on deep learning with class imbalance. J Big Data 6, 27 (2019). https://doi.org/10.1186/s40537-019-0192-5
22. K. Yamanishi, J.-I. Takeuchi, G. Williams, and P. Milne. "On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms," Data Mining and Knowledge Discovery, Vol. 8(3):275–300, 2004.

23. K.U. Jaseena, Binsu C. Kovoor, Deterministic weather forecasting models based on intelligent predictors: A survey, Journal of King Saud University - Computer and Information Sciences, 2020, doi: 10.1016/j.jksuci.2020.09.009.
24. Kothawale, D. & Revadekar, Jayashree & Kumar, K.. (2010). Recent trends in pre-monsoon daily temperature extremes over India. Journal of Earth System Science, 119(1), 51-65. Journal of Earth System Science. 119. 51-65. 10.1007/s12040-010-0008-7.
25. Liu Y, Zhao Q, Yao W, Ma X, Yao Y, Liu L. Short-term rainfall forecast model based on the improved BP-NN algorithm. Sci Rep. 2019 Dec 24;9(1):19751. doi: 10.1038/s41598-019-56452-5. PMID: 31875049; PMCID: PMC6930286.
26. M. Gupta, J. Gao, C.C. Aggarwal, J. Han 'Outlier detection for temporal data: A survey', IEEE Transactions on Knowledge and Data Engineering, 26 (9) (2014), pp. 2250-2267
27. Macroeconomic Nowcasting and Forecasting with Big Data, Brandyn Bok, Daniele Caratelli, Domenico Giannone, Argia M. Sbordone, Andrea Tambalotti Annual Review of Economics 2018 10:1, 615-643
28. Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2. Association for Computational Linguistics, 1003–1011.
29. Mohanakumar, K., Santosh, K. R., Mohanan, P., Vasudevan, K., Manoj, M. G., Samson, T. K., Kottayil, A., Rakesh, V., Rebello, R., & Abhilash, S. (2018). A versatile 205 MHz stratosphere-troposphere radar at Cochin - scientific applications. Current Science (00113891), 114(12), 2459–2466. https://doi.org/10.18520/cs/v114/i12/2459-2466
30. Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M. *et al.* Deep learning applications and challenges in big data analytics. *Journal of Big Data***2,** 1 (2015). https://doi.org/10.1186/s40537-014-0007-7
31. Parashar and P. Johri, "Short-Term Temperature and Rainfall Prediction at Local and Global Spatial Scale: A Review," 2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), 2021, pp. 742-746, doi: 10.1109/ICACITE51222.2021.9404767.
32. Pfülb B., Hardegen C., Gepperth A., Rieger S. (2019) A Study of Deep Learning for Network Traffic Data Forecasting. In: Tetko I., Kůrková V., Karpov P., Theis F. (eds) Artificial Neural Networks and Machine Learning – ICANN 2019: Text and Time Series. ICANN 2019. Lecture Notes in Computer Science, vol 11730. Springer, Cham. https://doi.org/10.1007/978-3-030-30490-4_40
33. Robert Nau, 'Notes on the random walk model', https://people.duke.edu/~rnau/411data.htm
34. S. Manandhar, S. Dev, Y. H. Lee, Y. S. Meng and S. Winkler, "A Data-Driven Approach for Accurate Rainfall Prediction," in IEEE Transactions on Geoscience and Remote Sensing, vol. 57, no. 11, pp. 9323-9331, Nov. 2019, doi: 10.1109/TGRS.2019.2926110.
35. S. Ramaswamy, R. Rastogi, and K. Shim. "Efficient algorithms for mining outliers from large data sets," in Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, vol. 29, pp. 427–438, Dallas, Texas, United States, May 16–18, 2000.

36. S. Zhao *et al*., "A Review of Single-Source Deep Unsupervised Visual Domain Adaptation," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 2, pp. 473-493, Feb. 2022, doi: 10.1109/TNNLS.2020.3028503.
37. Saidani, I., Ouni, A. & Mkaouer, M.W. Improving the prediction of continuous integration build failures using deep learning. *Autom Softw Eng***29,** 21 (2022). https://doi.org/10.1007/s10515-021-00319-5
38. Sánchez-Monedero J., Gutiérrez P.A., Pérez-Ortiz M., Hervás-Martínez C. (2013) An n-Spheres Based Synthetic Data Generator for Supervised Classification. In: Rojas I., Joya G., Gabestany J. (eds) Advances in Computational Intelligence. IWANN 2013. Lecture Notes in Computer Science, vol 7902. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-38679-4_62
39. Shaoke Wang, Zhaoyan Zhang, Peiguang Wang, Yaru Tian,Failure warning of gearbox for wind turbine based on 3σ-median criterion and NSET, Energy Reports,Volume 7, Supplement 7,2021,Pages 1182-1197, doi: 10.1016/j.egyr.2021.09.146.
40. Singhal, Richa & Rana, Rakesh. (2015). Chi-square test and its application in hypothesis testing. Journal of the Practice of Cardiovascular Sciences. 1. 10.4103/2395-5414.157577.
41. Stephen H. Bach, Bryan He, Alexander Ratner, and Christopher Ré. 2017. Learning the Structure of Generative Models without Labeled Data. In International Conference on Machine Learning (ICML).
42. Stephen H. Bach, Daniel Rodriguez, Yintao Liu, Chong Luo, Haidong Shao, Cassandra Xia, Souvik Sen, Alex Ratner, Braden Hancock, Houman Alborzi, Rahul Kuchhal, Chris Ré, and Rob Malkin. 2019. Snorkel DryBell: A Case Study in Deploying Weak Supervision at Industrial Scale. In Proceedings of the 2019 International Conference on Management of Data (SIGMOD '19). Association for Computing Machinery, New York, NY, USA, 362–375. DOI: /10.1145/3299869.3314036
43. Thara Anna Mathew, Neelam Malap, M.G. Manoj, Y. Jayarao, Kiran Todekar, V. Rakesh, Rejoy Rebello, K. Mohankumar, Thara Prabhakaran, 'Pre-monsoon convective events and thermodynamic features of southwest monsoon onset over Kerala, India – A case study', Atmospheric Research, Volume 248,2021,105218, DOI: 10.1016/j.atmosres.2020.105218. thermodynamic variables and convective precipitation. J. Appl. Meteorol. 38,
44. Truong Son Pham, Quang Uy Nguyen, and Xuan Hoai Nguyen. 2014. Generating Artificial Attack Data for Intrusion Detection Using Machine Learning. In Proceedings of the Fifth Symposium on Information and Communication Technology (SoICT '14). ACM, New York, NY, USA, 286–291. https://doi.org/10.1145/2676585.2676618
45. Véronne Yepmo, Grégory Smits, Olivier Pivert, Anomaly explanation: A review,Data & Knowledge Engineering, Vol 137,2022,Doi: 10.1016/j.datak.2021.101946.
46. W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan. 2004. Using artificial anomalies to detect unknown and known network intrusions. Knowl. Inf. Syst. 6, 5 (September 2004), 507–527. DOI: 10.1007/s10115-003-0132-7
47. Wang, S., Minku, L.L., Chawla, N. and Yao, X. (2019) 'Learning from data streams and class imbalance', Connection Science, Vol. 31, No. 2, pp.103–104, DOI: 10.1080/ 09540091.2019.1572975.

48. Weiping Diao, Ijaz Haider Naqvi, Michael Pecht, Early detection of anomalous degradation behavior in lithium-ion batteries, Journal of Energy Storage, Volume 32,2020, doi: 10.1016/j.est.2020.101710.
49. Wenlong Chen, Xiaoling Wang, Dawei Tong, Zhijian Cai, Yushan Zhu, Changxin Liu,Dynamic early-warning model of dam deformation based on deep learning and fusion of spatiotemporal features, Knowledge-Based Systems, Volume 233,2021,107537, doi: 10.1016/j.knosys.2021.107537.
50. Xue Xu, Jinliang Ding, 'Similarity and sparsity collaborative embedding and its application to robust process monitoring, Control Engineering Practice, Vol 122,2022, doi: 10.1016/j.conengprac.2022.105113.
51. Yipeng Wu, Shuming Liu, Xue Wu, Youfei Liu, Yisheng Guan, Burst detection in district metering areas using a data driven clustering algorithm, Water Research, Volume 100,2016, pp 28-37, doi: 10.1016/j.watres.2016.05.016.
52. Yuke Zeng, Huanxin Chen, Chengliang Xu, Yahao Cheng, Qijian Gong,A hybrid deep forest approach for outlier detection and fault diagnosis of variable refrigerant flow system, International Journal of Refrigeration, Volume 120,2020,Pages 104-118,Doi:/10.1016/j.ijrefrig.2020.08.014.
53. Zawadzki, I., Torlaschi, E., & Sauvageau, R. (1981). The Relationship between Mesoscale Thermodynamic Variables and Convective Precipitation, Journal of Atmospheric Sciences, 38(8), 1535-1540. Retrieved Mar 4, 2022, from https://journals.ametsoc.org/view/journals/atsc/38/8/1520-0469_1981_038_1535_trbmtv_2_0_co_2.xml
54. Zhifeng Jia, Zilong Guan, Zhao Liu & Dongming Yang (2020) Influence of short-term rainfall forecast error on flood forecast operation: A risk assessment based on Bayesian theory, Human and Ecological Risk Assessment: An International Journal, 26:9, 2447-2461, DOI: 10.1080/10807039.2020.1768360
55. Zulkipli, N & Satari, Siti & Yusoff, W. (2021). A synthetic data generation procedure for univariate circular data with various outliers scenarios using Python programming language. Journal of Physics: Conference Series. 1988. 012111. 10.1088/1742-6596/1988/1/012111.

Figures

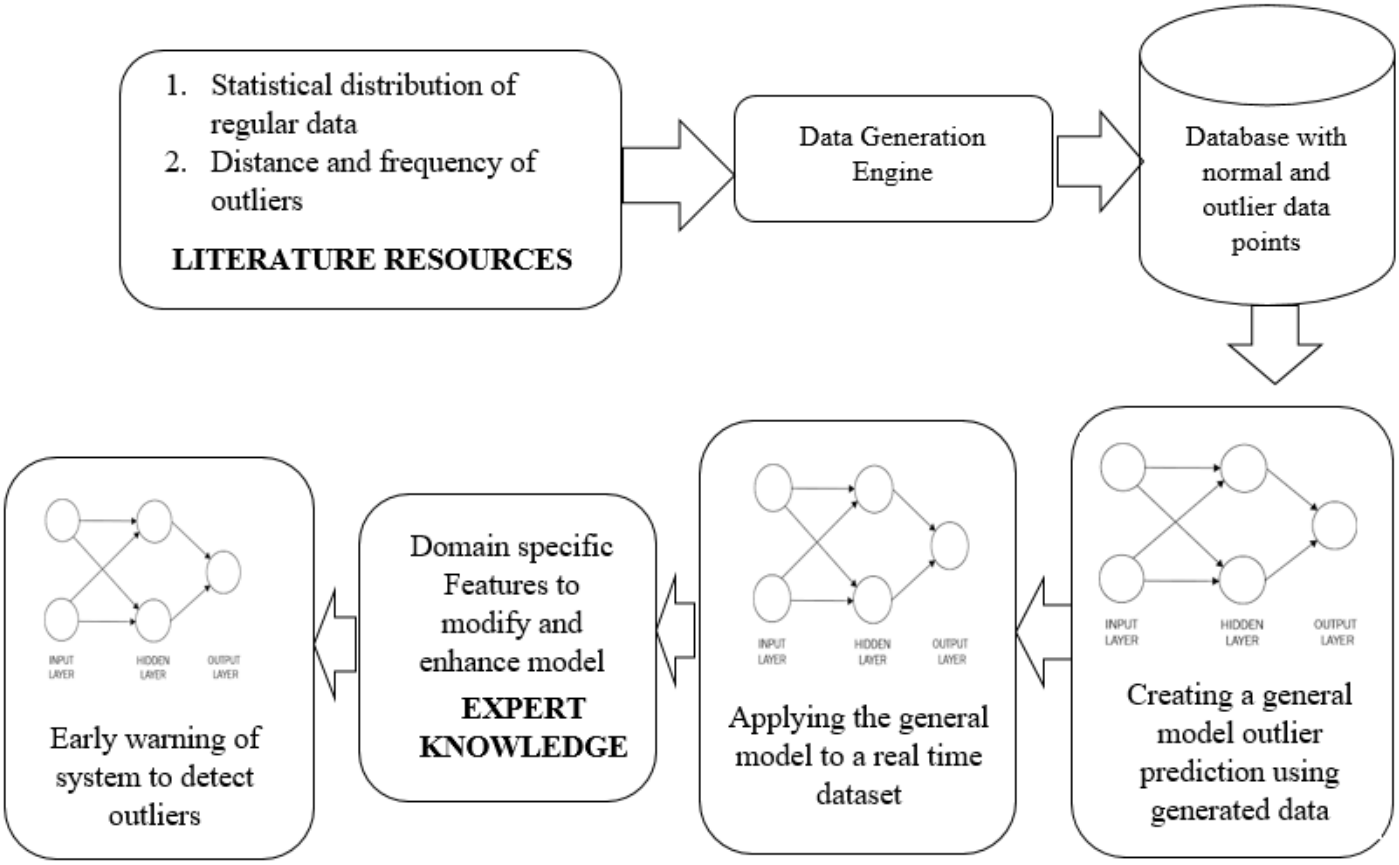

Figure 1

Proposed system architecture

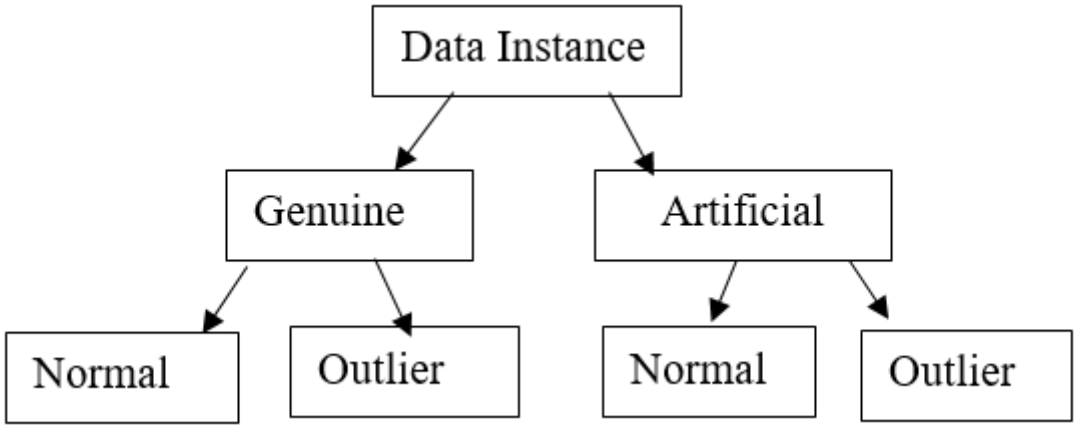

Figure 2

Terminology of data instances

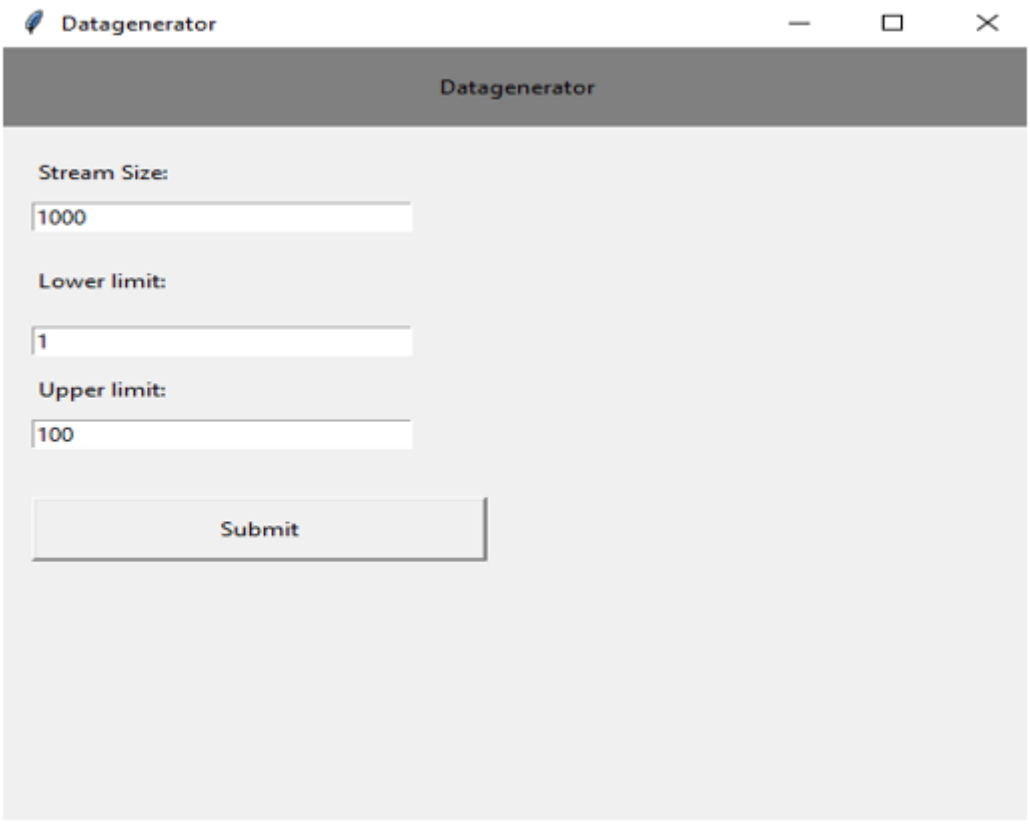

Figure 3

Data Generator User Interface

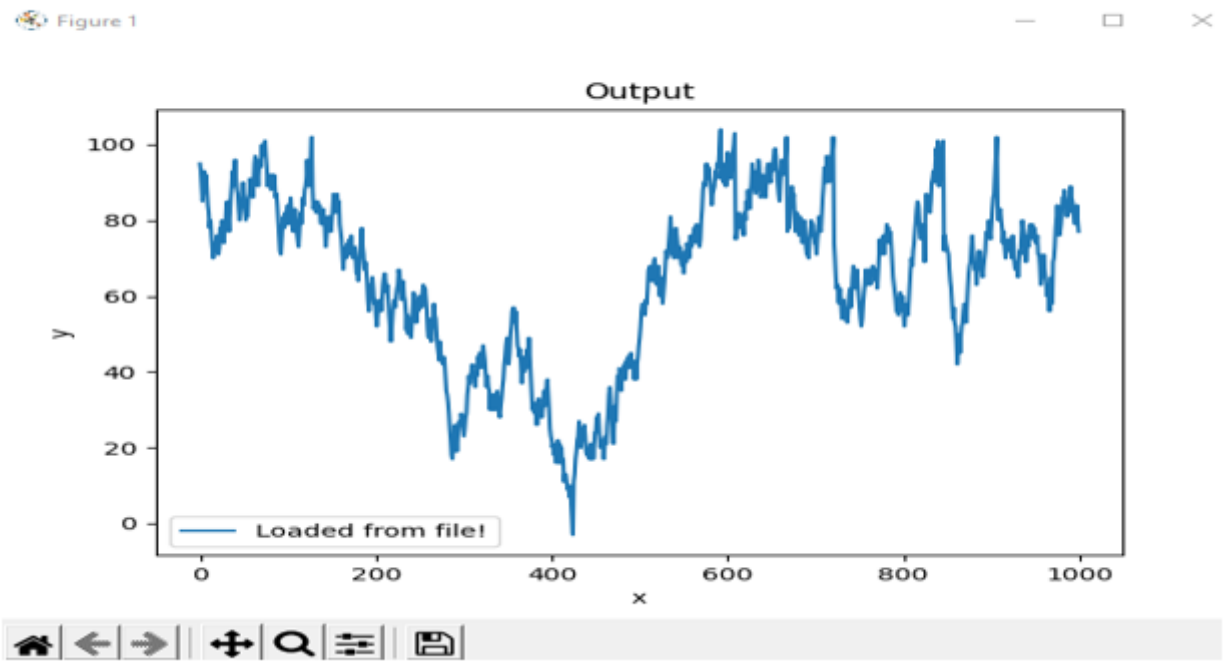

Figure 4

Dataset retrieved from the database

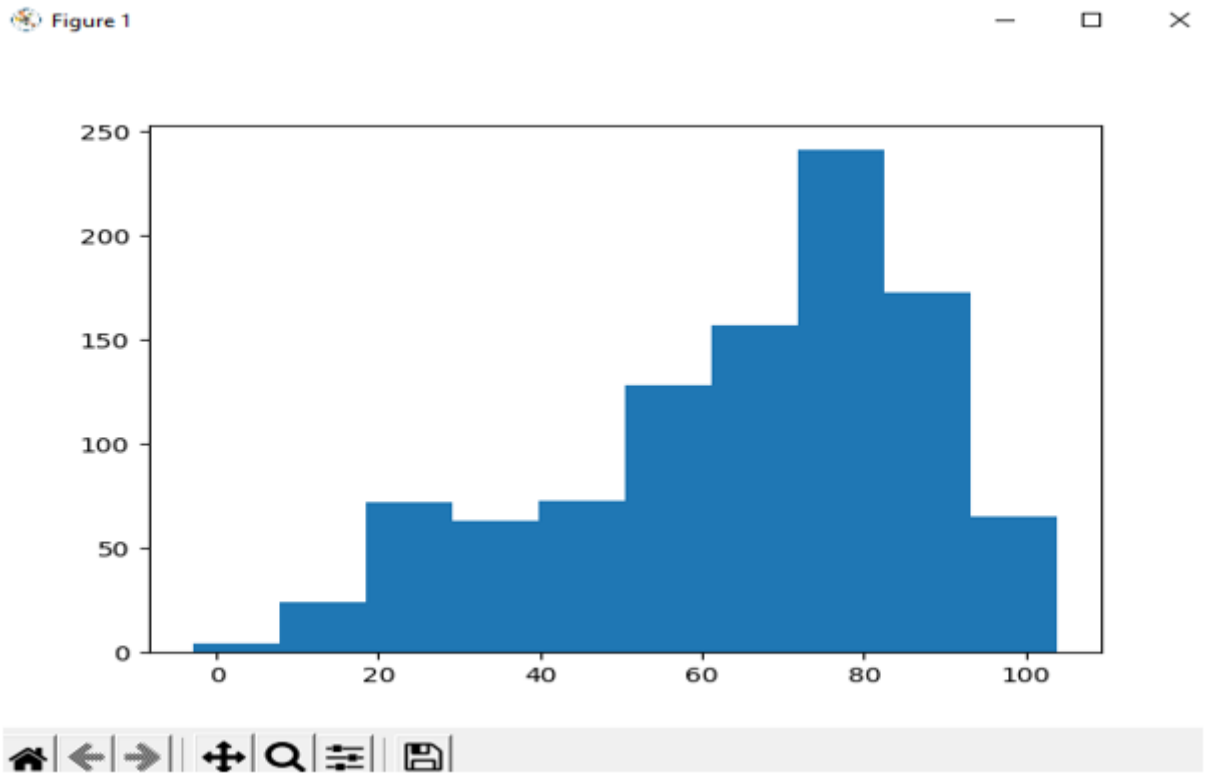

Figure 5

Histogram of generated dataset

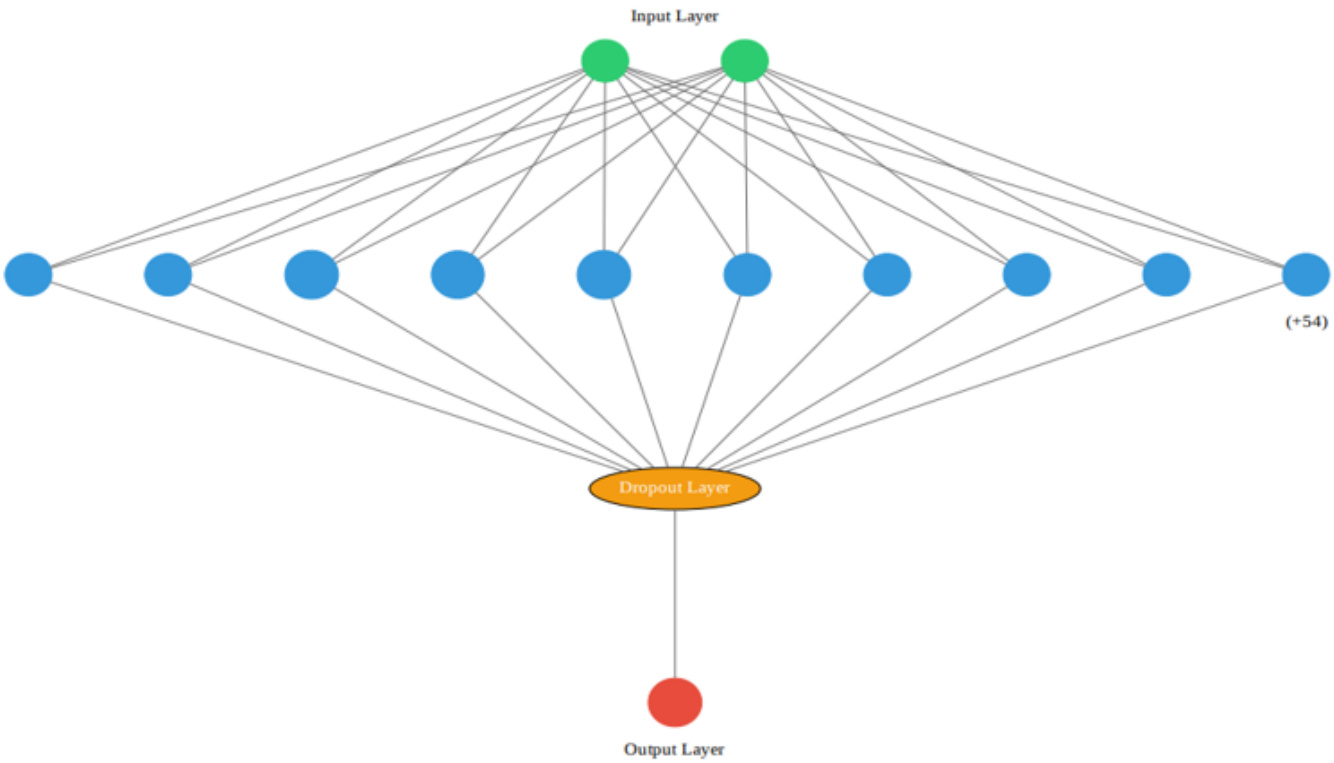

Figure 6

Structure of base neural network

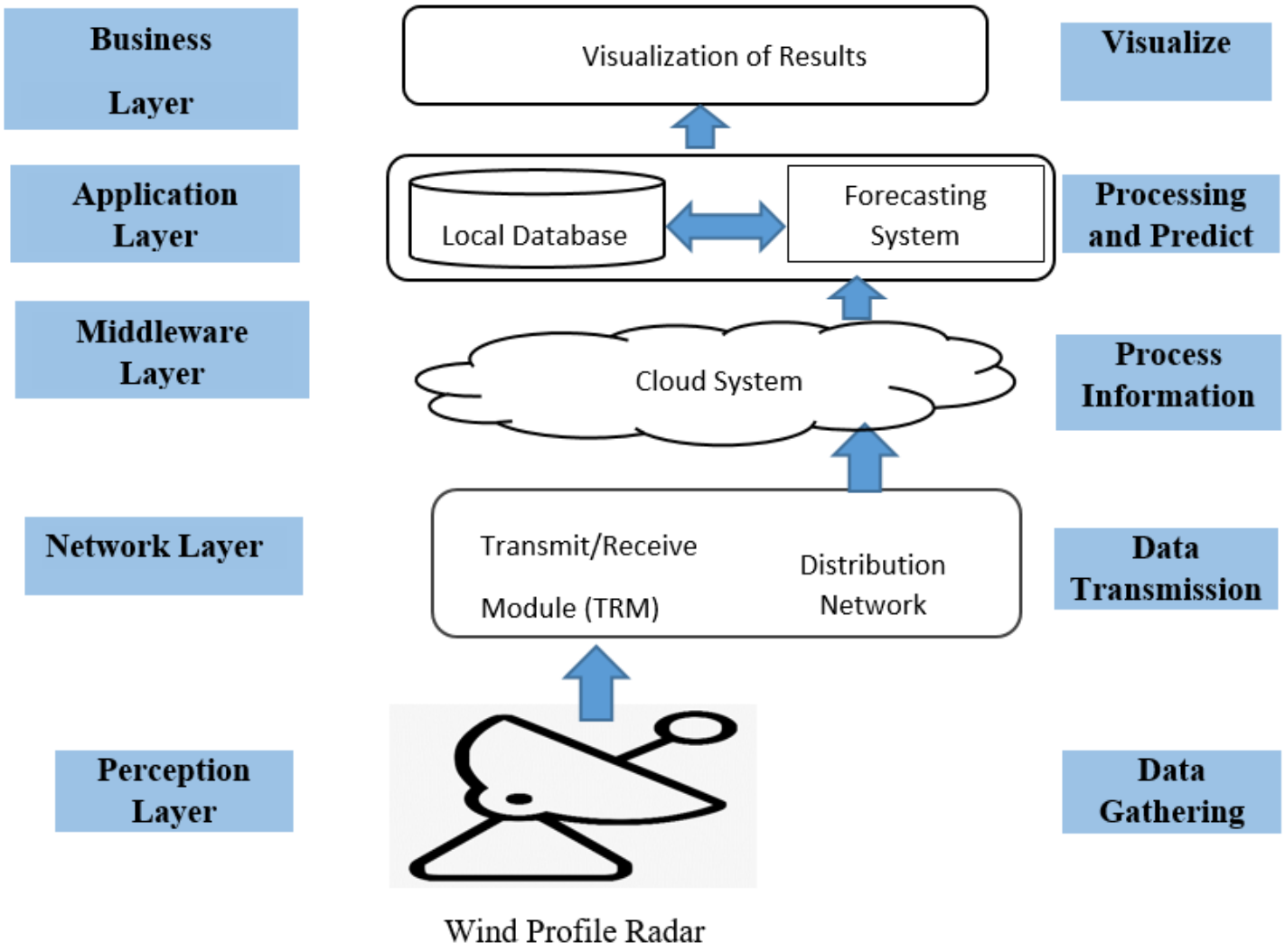

Figure 7

Architecture of the data processing system

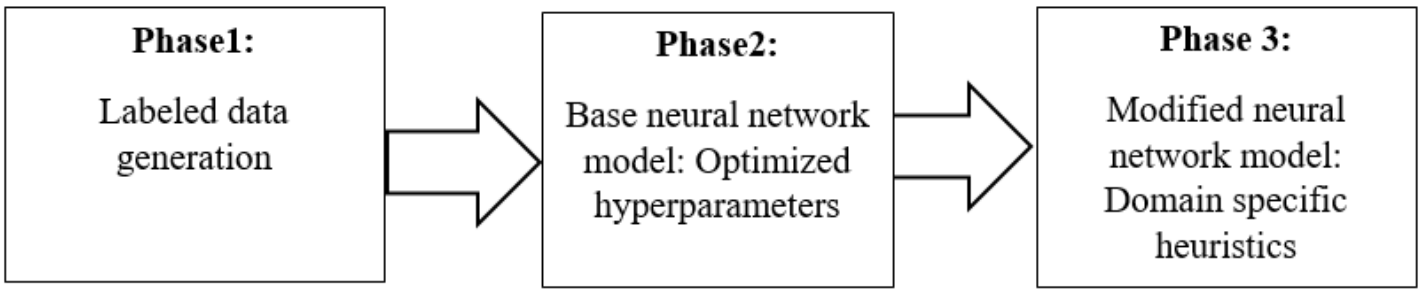

Figure 8

General framework of current study