Anomoly Detection in ECONet Dataset

Vishnu Challa
vchalla2@ncsu.edu

Sujith Tumma
stumma2@ncsu.edu

Indu Chenchala
ichench@ncsu.edu

Prashan Sengo
psengo@ncsu.edu

1. INTRODUCTION AND BACKGROUND

1.1 Problem Statement

Within North Carolina State University the North Carolina State Climate Office has a project called ECONet(Environmental Climate Observing Network) that takes a variety of different measures ranging from air temperature to soil moisture. This is done for the express purpose of improving weather data, predicting severe weather situations and providing publicly accessible weather data to the world. To do this ECONet has 43 stations that record sensor data across North Carolina at 1 minute intervals resulting in more than 1 million plus data points being recorded every year. With this large amount of data it is easy for erroneous values to be missed and manually having people go through the dataset to detect erroneous values is infeasible. As such our research on Anomaly detection through improved imbalanced sampling techniques could be very useful to solving this issue.

The main issue with the identification of erroneous values in the ECONet data is the imbalanced nature of the erroneous values. About 97 percent of the data collect by ECONet stations are found to be non erroneous whereas the rest 3 percent have errors or anomalies in one of the many sensor data. This means that the standard Machine learning approach of training a classifier using the data and then predicting which values are anomalous may not be as effective as one would think. This is seen in the example of having a classifier that always predicts values as non erroneous which would result in an accuracy of about 97 percent as it only miss-classifies the 3 percent of erroneous data. This would not properly reflect the true performance of the model and as such in order to rectify this problem caused by the imbalance of values we will be taking a look at several unique approaches.

In general we have two main approaches we will be looking at the first is using a sampling technique that samples in a stratified manner to ensure the dataset is comprised of an even number of erroneous and non erroneous values. For the sampling techniques we will be looking at SMOTE, Under-sampling and Oversampling which should help make the data set not imbalanced. We will also compare the performance of doing these techniques on several baseline classifiers(Bayes net, Random Forest, KNN, and LSTM) that didn't have an imbalanced sampling technique done to them in order to see if an effective performance increase was seen.

1.2 Related Work

The first reference we have referred is a paper on ECONet dataset[2]. This is to understand the data well before we proceed with the implementation part. As an outcome of this reading we have understood how some of the attributes like Range check(R flag), Buddy check(B flag), Intersensor check(I flag) and Trend check(Z flag) are being calculated. We have also understood how temperature values for each of stations are being recorded timely basis. All this information helped in performing EDA and preprocessing on the training data to get some useful insights out of it. And also as an outcome of this paper we have decided to perform time series analysis on the data using LSTM to further observe the changes in the temperature trends of each station in particular time intervals of an year.

One of our references to deal with imbalanced data is a paper published by Mohamed Bekkar, Dr.Hassiba Kheliouane Djemaa and Dr.Taklit Akrouf Alitouche[1]. This paper primarily talks about the important set of metrics that needs to be considered while dealing with highly imbalanced data. Most of the paragraphs explain some important terms in model evaluation like Precision, Recall(Sensitivity), Specificity, Accuracy, Error rate, Confusion matrix, G-Means, F-Measure, Balanced accuracy, Matthews correlation coefficient, Precision-Recall curve and ROC curves and how they can be used to handle imbalanced data. In overall it gives us a rough idea on all the required attributes to evaluate our model. As a conclusion from this reference we have decided to give more weight to Recall, F-Measure(specifically F-2 Score) and area under Precision-Recall curve while evaluating our model against imbalanced data. We will also maintain the values of remaining metrics like Precision, Specificity and Accuracy at a balanced level but will not bother on achieving them to the fullest.

In the field of techniques for anomaly detection there are plenty of sources and research papers that are available out in the world. One such research paper is the paper by Lei Wang on classification model on big data in medical diagnosis based on semi supervised learning[3]. This paper in particular talks about using Semi-Supervised learning to classify unlabeled medical data and comparing this performance to a standard supervised learning approach. The main focus that we looked at was on the methodology of using semi-supervised learning. In particular the paper delineated that a supervised algorithm was original trained on a labeled dataset and then was made to predict the labels of the unlabeled dataset. The supervised learning dataset then trains on the original labeled dataset and the unlabeled dataset that is now labeled. Doing so was shown to result in a increase in classification accuracy. The method in the paper could be useful in taking advantage of the unlabeled extraneous ECONet data that was given and could potentially increase the amount of erroneous data points we have available resulting in a less imbalanced dataset.

2. METHOD

2.1 Approach

As a part of data preprocessing and EDA, we have performed the label-encoding on some of the non-numeric attributes and also did z-score normalization on the value measures to prevent data overflow. We have also plotted some visualizations to observe the changes in the temperature trends on a time basis for each station. This helped us to understand the meaning and correlation between data attributes.

Within our investigation of the classification models we have discovered that the models can be placed in 3 different categories: linear, non-linear and ensemble. Linear models are models whose outputs are a mathematical combination of the inputs. For instance a regression models output is a weighted linear combination of the inputs. The non linear models are models whose outputs are not based on some mathematical combination of the inputs. The final type of models we have is the ensemble models which is basically many different models that aggregate their classification results. From these categories we chose to explore KNN and Decision Tree for non linear models, Naive Bayes for linear models and random forest for ensemble type models.

1. Random Forrest Classifier (RFC)
   Another supervised machine learning technique is the Random Forest Classifier, which involves training several uncorrelated Decision Trees on a dataset in such a way that their combined predictions outperform any single tree's prediction. The most significant feature of a Random Forest Classifier is that the different constituent Decision Trees should have no association. A Random Forest Classifier, on the other hand, employs the bagging technique, in which a different random subset of data is chosen at each step to train the individual Decision Trees. The use of RFCs was mentioned for handling imbalanced data in several more works that we discovered throughout our references. Since we have highly uncorrelated data in our sample and our goal is to predict erroneous records, we went ahead and attempted to classify our dataset using RFCs. We will go into the specifics of how to use this approach in the further sections.

2. Naive Bayes Classifier
   Naive Bayes Classifier is a supervised learning algorithm that uses the Naive Bayes theorem and the assumption of independence among the predictors. Naive Bayes assigns and updates probabilities of each of the features given the target and uses these probabilities to calculate how likely a certain combination of features result in a particular value for the target. Naive Bayes was chosen for this problem due to the presence of categorical variables in the dataset. Since Naive Bayes can handle the presence of categorical variables as it is getting the probability of the categorical variable rather than using its actual value(the actual value would be a string not number) it was chosen over other linear models that can't handle categorical input.

3. Decision Tree Classifier
   Decision Tree is a Supervised Machine Learning Algorithm that makes judgments based on a set of rules, similar to how people do.The goal is to learn simple decision rules from data attributes to develop a model that predicts the value of a target variable.On imbalanced data, decision trees usually perform well. They work by learning a hierarchy of if/else questions, which forces them to handle both classes.Because our dataset contains outliers, We investigated using a decision tree classifier. Outliers have a lesser impact on decision tree data.

4. KNN
   KNN is a supervised machine learning algorithm that relies on labeled input data to learn a function that produces an appropriate output when given new unlabeled data. Both classification and regression predicting problems can be solved with KNN. However, it is more commonly employed in classification problems. With KNN ,The output is simple to interpret, takes less time to calculate and has good predictive power. And also it is a lazy learner with zero effect on the performance even on adding new records. Since the weather data keeps varying on a timely basis and satisfies the above characteristics, we have chosen to try out this model.

2.2 Rationale

1. Random Forest Classifier
   All the attributes we have in our input are uncorrelated. Even the flag checks have a unique significance. Due to this we have taken RFC into consideration which performs best with the uncorrelated data. RFC is an excellent classification technique that is very simple to use. It is resistant to over-fitting the dataset since it employs a bagging technique to blend the output from numerous independent decision trees. Due to the randomized approach employed in bagging for training purposes, the individual decision trees formed are also uncorrelated. As a result, a classifier with good training and test accuracy has been created. This fact is also demonstrated in the experiments presented in the next sections. RFC, when compared to other methods, would theoretically outperform because it does not over fit the dataset. In addition, being an ensemble method, it offers better test accuracy than the SVM, for example.

2. Naive Bayes Classifier

We chose Naive Bayes over another linear model like linear regression because Naive Bayes can handle categorical input compared to linear regression which can't process the non numeric value of the categorical variable. One concern with running Naive Bayes is if any categorical variable has a value that doesn't get observed in the training data we would see that the probability for that value would be 0 and won't be able to assign a prediction. In order to handle this a smoothing parameter was added which adds a very small probability to all the probabilities ensuring that there are no 0 probabilities.

3. Decision Tree Classifier
   As the Decision tree classifier need not require any data preparation and cleaning, we have chosen it as a baseline model to see how the other similar classification models outperform based on our baseline evaluation metrics. And also the Output of the decision tree model is easy to interpret with no perquisite knowledge on statistics like the other models.

4. KNN
   When a new unlabeled instance is provided to KNN, it does not fit on the train data and instead looks for the nearest instance. As a result, we may infer that adding more training data will have no effect on the model's performance which might not be the case for the other similar models. For example if we consider the classifiers like Adaboost and SVM on adding new instances to the train data, model's performance might get affected. Moreover in our case, Since there might be instances where we need to add additional records to the train data as it varies timely basis, we have chosen to experiment on this model.

3. EXPERIMENT

3.1 Dataset

The dataset used in this project is ECONet.It has come from NC State's Climate office where the data is collected from 43 stations that record several sensor data from across North Carolina. Attributes include : Station, Ob, measure, target, R_flag, I_flag, Z_flag, B_flag. The data contains train and test files. Also some extra files to make more sense of the data. This information is used by our model to determine if the predicted measure values are erroneous or not. There are around 96.5 percent of instances of the false class and only 3.5 percent of true class in the train data.

In terms of how we split the dataset for training purposes, we used a 70 / 30 split for training and validation sets respectively. So 70 percent of a dataset considered as a training set (4615292 data points) and 30 percent of a dataset considered as a validation Set (1977982 data points) for most of the baseline models.

3.2 Hypothesis

The primary hypothesis we are looking to answer is whether we can improve the detection of erroneous values within the ECONet dataset. In particular we are hoping to explore whether using the imbalanced data sampling techniques like SMOTE, undersampling, and oversampling results in an improved performance of the detection of anomolous values. We will also explore whether using a model like LSTM that takes into account the time series nature of the data will result in an improved performance over the base line models.

3.3 Experimental Design

- Feature transformation and Sampling techniques
  As a part of preprocessing step we have applied techniques like feature transformation and data sampling. We have applied z-score normalization on the value measure to avoid data overflow issues. And also transformed the non-numeric attributes using label encoders so that it will be easier to process them while training. We have performed both under sampling and over sampling using SMOTE to maintain equal distributions of each classes.

- Data Cleaning and Selection (check for null values)
  In the first stage of preprocessing we checked if there are any null values in the dataset but we figured out there are no null values in the dataset.

- Model Training (cross validation, hyper parameter selection)
  For each of the four classifiers we are looking at(KNN, Naive Bayes, Random Forest, and Decision Tree) we have chosen the best hyper parameters by running grid search with a combination of specified parameters and selected the parameter combination that results in the best f1-score. Grid search takes a list of parameters that the model can have and runs 10 fold cross validation varying the parameters and returns the combination of parameters that has the best specified scoring metric. The following parameter and parameter values we have tried is listed below:
  - Random Forest [n_estimators: [10, 20, 30, 40, 50, 60, 70, 80, 90 100], criterion: [”gini”, ”entropy”]]
  - Naive Bayes [var smoothing: [1e-9, 1e-8, 1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1]]
  - KNN [k:[0 to 31]
  - Decision Tree[max_depth: [1,3,5,7,9,10,11,15], criterion:[”gini”,”entropy”]]

- Evaluation Metrics (recall, precision, etc.)
  For all baseline models we used 70 percent data as the train data and 30 percent of the data as the test data.

  We have considered recall,Fscore,Fbeta score,precision-recall curve as the metrics to evaluate the models.

4. RESULTS

For each of the classifiers we have produced the metrics from the metric section in the experimental design section along with a short evaluation of the results for each classifier.

Classifier	Accuracy	Precision	Recall	F1-score	F2-score
Random Forest	0.99	0.99	1	0.99	0.99
Naive Bayes	0.98	0.0	0.0	0.0	0.0
SMOTE - KNN	0.96	0.99	0.95	0.95	0.95
Decision Tree	0.995	0.95	0.90	0.92	0.91

Looking at the results a few observations can be seen for each of the results for the different classifers:

- Random Forest Classifier: For random forest classifier we have performed a grid search by changing parameters like n_estimators and criterion. And also we have performed multiple sampling techniques like oversampling using SMOTE and a combination of both undersampling for the majority class and oversampling using SMOTE for the minority class to observe the results. Base on the results we have observed that the sampling techniques mentioned above did not bring any significance difference to the results. And the area under precision-recall curve was a perfect 1. As per our understanding this is due to the majority voting strategy followed by random forest classifier decision trees used to predict outcome while training the dataset.

- Naive Bayes: For Naive Bayes we see that everything except accuracy is 0. On furthur investigation we have seen when outputting a confusion matrix of the predicted and true target values of the classifier's results that the TP and FP were both 0 causing the recall and precision to be 0. On Further investigation it was seen that this value occurred because the Naive Bayes model was always predicting that there was no erroneous values resulting in a very poor precision, recall, and f-scores. The high accuracy occurred because 98 percent of the data is non erroneous and since we always predict non erroneous we correctly predict all the data except the 2 percent of erroneous data.

- Decision tree classifier: By Performing the Grid Search CV with the hyper-parameters like max-depth and criterion gini for different values we obtained the above results. On plotting the precision-recall curve the area under curve was 0.93. We have considered these result as a baseline standard for the other models.

- KNN: For KNN using k=3 we obtained the above results. On plotting the precsion-recall curve the area under curve was 0.961. We tried on basic KNN classifier and got low performance and tried SMOTE KNN and got good results which can be seen in the above table.Also, the above hyper parameter of k is derived from performing GridSearchCV.

5. PROPOSED WORK

5.1 Design of Future Experiments

We've made progress so far in normalizing the data,encoding the data and applying a few Machine Learning algorithms to gain a sense of the classification problem we're dealing with. We could improve on this initial work by applying time series analysis in LSTM Model determining whether or not we could improve the precision and recall comparetively more than the baseline models that we have trained. Also by applying the sampling techniques of imbalanced data on the LSTM model.

1. **Explore Sampling techniques for imbalanced dataset using LSTM**
   We think exploring the sampling techniques for imbalanced dataset like SMOTE,under sampling,over sampling on LSTM model might give better results like the baseline models and see if we could get good results comparatively.

2. **Explore Time Series analysis in LSTM model**
   The algorithms that we have tried on our dataset till now include Classifiers Random Forest,Decision Tree, KNN and Naive Bayes .As our dataset is timeseries dataset .We believe we could experiement with LSTM model which works best for time series data.LSTM is a special kind of RNN.It is capable of learning long term dependencies.Remembering information for long period of time is particularly their default behavior. It is Widely used where the data is sequential that is with respect to time the value of attribute changes.Best application is time series data.With the LSTM the data is trained sequentially .In case of base line models there is no guarantee that data is trained in a sequential manner.This may decrease the performance of the model.So, see if LSTM may outperform the baseline models that are considered.

3. **Model Evaluation and Hyperparameter Tuning of LSTM**
   In LSTM there are many hyper parameters like number of layers,number of nodes in each layer ,learning rate,activation function etc.We intend to spend time in figuring out and experimenting with these hyperparameters that could help us in improving the models performance

6. PLAN OF ACTIVITIES

Tasks	
TaskName	**Person**
Run models with imbalance sampling techniques	Vishnu
Take data from full folder and put in correct format	Prashan
Create method to grab past 200 rows from given Date	Indu
Create LSTM model	Sujith , Prashan
Tune hyperparams of LSTM	Vishnu , Indu
Finish research paper write up	Everyone

6.1 Online Collaboration

- March 17 2022: 9:00PM - 10:30PM
- March 18 2022: 9:00PM - 10:30PM
- March 19 2022: 9:00PM - 10:30PM
- March 20 2022: 9:00PM - 10:30PM
- March 21 2022: 9:00PM - 10:30PM
- March 20 2022: 9:00PM - 10:30PM
- March 28 2022: 9:00PM - 10:30PM
- March 30 2022: 9:00PM - 10:30PM
- April 7 2022: 9:00PM - 10:30PM
- April 8 2022: 9:00PM - 10:30PM
- April 11 2022: 9:00PM - 10:30PM

7. REFERENCES

[1] Dr.Taklit Akrouf Alitouche Mohamed Bekkar, Dr.Hassiba Kheliouane Djemaa. Evaluation measures for models assessment over imbalanced data sets. URL: https://tinyurl.com/4fwvjtcm.

[2] John A. McGuire Heather A. Dinon Aldrige Ryan P. Boyles Sean P. Heuser*, Aaron P. Sims. Quality control methods for the north carolina environment and climate observing network. URL: https://tinyurl.com/3p6y2ku5.

[3] Lei Wang, Qing Qian, Qiang Zhang, Jishuai Wang, Wenbo Cheng, and Wei Yan. Classification model on big data in medical diagnosis based on semi-supervised learning. *The British Computer Society*, 2020. URL: https://tinyurl.com/yckwsxcy.