Anomaly Detection in weather data exchanging IoT devices

Saadat Hasan Khan
Department of Computer Science
George Mason University
Fairfax, USA
skhan225@gmu.edu

***Abstract*—One of the technologies with the fastest global growth is the Internet of Things. Since IPv6 has stabilized, there are many open addressing spaces that permit even sensors to interact with one another while gathering data, let alone vehicles that communicate while moving. The Internet of Things (IoT) has altered how machines, objects, and people interact with one another. Security and privacy concerns are, however, a very serious matter as a result of its expansion. There is a good chance that the network may contain aberrant values that give a deceptive portrayal of the true situation because so many devices exchange data over the internet. As the goal of any issue solver would strongly depend on the data, such abnormal numbers can potentially result in mistakes when employing these data for work. In such circumstances, the communication network's security should be powerful enough to detect suspicious data. IoT devices that exchange weather data are a prime example of such a situation. If anomalous data are transmitted through the network used to exchange weather data, the weather conditions may be misrepresented. Using a dataset that is most appropriate for this goal, we present two machine learning algorithms that can be utilized to recognize any aberrant data in an IoT that exchanges meteorological data. Finally, statistical tests are conducted to determine which algorithm performs the best after statistical comparisons between these models are completed.**

***Index Terms*—IoT (Internet of Things), Anomaly detection, Machine learning algorithms, Statistical Analysis.**

I. Introduction

Internet of Things (IoT) is a hot field, capable to alter our lives in various ways [1]. However, they rely on sensors-which could have high or low accuracy. These sensors gather data of the surroundings it is in. Sensors behave by either perceiving data only or by perceiving the data and acting on that environment using the data based on some decisions. The extent to which IoT impact our lifestyles is huge. They help people in making decisions too. For example, a health supervision system uses high accuracy sensors to analyse the person's heart data. Using that data, the system may be able to book an appointment with the doctor if an emergency check-up is needed. These smart devices connected to the Internet are capable of generating data on its own based on behavior or expected result and share it over the Internet. This is what constitutes the concept of Internet of Things (IoT).

But as IoT expanded, damaging assaults became more prevalent [2]. According to Gartner's Research [3], there will be 26 billion IoT devices by the end of 2026. Logical controls like software shields are used to make sure that the system and data are accessed by the authorized individuals - which use passwords, access control logical controls, etc [4]. A step for preventing IoT attacks is implementing an Intrusion Detection System (IDS). IDSs should be able to figure out that it is under an attack during the time of an intrusion. Furthermore, it should have the capability to not just understand when a system is under attack, but also act on it (Gordon, 2015). It could be done by sending alerts or notifications to the users. There are two ways an intrusion detection system works basically: 1) Signature based detection 2) Anomaly based detection. Signature based attacks work very well when the attacks are previously analysed. They are generally rule based systems. Anomaly based detection are useful against unknown attacks or security breaches [5]. Anomaly detection techniques heavily rely on AI based techniques and using Machine Learning tools. The general framework is to make the subject learn from the data. The AI models or Machine Learning tools learn which data is theirs and which are anomalous and then separate the anomalous data from the general data. Conversely, to detect specific patterns of the attacks that are known by investigating network data or traffic, signature based detection methods are powerful and have been more successful [6].

II. Literature Review

A lot of work has been done IoT's anomaly detection. However, a large threshold of these works focus on network intrusion detection. Their main focus is on machine learning algorithms like regression, SVM, bagging, boosting algorithm sets perform and their comparisons. [7] is another paper that provides insights on the performance of learning techniques. It experiments with the existing network intrusion detection tools and finds the best solution.

A big challenge for intrusion detection in IoT devices is obtaining labeled data. KDD (Knowledge Discovery and Data Mining) Cup 1999 dataset [8] published by MoD is one of the free available datasets but after a long time it has been declared as obsolete as well. [9].

The authors of the paper [10] points out differences in host and network-based intrusion detection techniques and

explains how they can provide better intrusion detection and protection against anomalous data [10]. The main goal is to examine the potential impact of a selection of ML techniques in order to build robust defenses for IoT environments based on the dataset for the subsequent two layers: I. Application Layer (Host based) II. Network Layer (Network Based) The application layer processes input from the system and produces an output using machine learning, data mining, data processing, and other analytics. The research "A Model for Anomalies Detection in Internet of Things (IoT) Using Inverse Weight Clustering and Decision Tree" utilizes a freely accessible unlabeled IoT dataset from Intel Berkeley Research Lab to work on the intrusion detection at application layer [11]. More than 2.3 million signals from 54 sensors in the Intel Berkeley Research facility were captured in the dataset in 2004 [12]. The dataset has to be processed because its primary purpose is anomaly detection research rather than intrusion detection. In the paper [13], machine learning methods are used to attack IoT devices. For their dataset, they also used methods like decision trees.

There aren't enough appropriate datasets available, despite the fact that cyber-security research is practically at its height. Recently, the UNSW NB15 dataset [14], which is IoT-eligible, has saved the day for contemporary intrusion-based network layer research. This dataset contains a sizable number of instances of both legitimate and malicious network traffic. It combines recent assault cases with typical behavior in traffic. For many common, modern attack methods, malicious traffic is produced. The UNSW-NB15 dataset is uneven because there are fewer instances of harmful attacks than there are of the frequent normal traffic. This dataset is giving a lot of networking IDS studies good support [15]. Several supervised learning methods, including SVM (Support Vector Machine) and a variety of ensemble classifiers, have been used and their performances compared by academics in the paper [9]. The ensemble methods integrate fundamental classifiers that have been trained on several dataset subsets. Based on an efficient classification model constructed inside the IDS, the IDS (Intrusion Detection System) is capable of differentiating between abnormal behaviors and normal behaviors. An IDS is built on SVM (Support Vector Machine) and C5.0, a popular implementation of a decision tree, aiming to deliver a response with reference to the specific attributes in the input record. Due to its lack of familiarity with such novel threats, existing IDS typically fails to identify attacks in key situations. Data may be split into classes using C5.0 and SVM, and after training, SVM will be able to map the data into a high-dimensional space [16].

III. Problem Description and Research Tools used

A. Description of the problem

Detecting anomalous values in a weather based IoT system can be tricky as anomalous values can come in different forms and types. Therefore as previously discussed, it has already been asserted that using machine learning methods can be much more useful than rule based systems [11]. Here, we are using the Intel Berkeley research dataset [12] for our research. The dataset contains weather data which contains anomalous values. We will use this data to train 2 different Machine Learning problems and see how well they perform. After that, we will carry out statistical analysis to compare these two machine learning models in terms of performance. We will show that they are significantly different and that one model is significantly better than the other for this specific problem.

B. Research Tools used

The research used a lot of techniques and tools in order to be whole. Most of the tools are statistical or mathematical. The following list provides them. The explanation and methodology of how they have been applied to our research is explained in the Research Methodology section.

Tools used for research:

- Sampling from Data.
- Feature selection
- Find Mean, Median, Quartiles, Box and Scatter Plots of the data
- K Means Clustering Algorithm for clustering Data
- Naive Bayes Classifier and Decision Tree Classifier to classify anomalous/normal data
- Hypothesis Testing
- Calculations using Confidence Intervals
- ANOVA to check whether the models are significantly different
- Performing Approximate Visual Test to compare the models
- Paired Observations to compare the models

IV. Research Methodology

There are many folds to the methodology used for this research. The steps are as follows:

A. Dataset

The Intel lab data was used which contained around 2.3 million of data using 54 sensors in the Intel Berkeley lab. Every 31 seconds, Mica2Dot sensors equipped with weather boards gathered time-stamped topology data along with measurements of the humidity, temperature, light, and voltage. Data was gathered using the TinyOS-based TinyDB in-network query processing technology. The data contained many features but only 4 features represented they environmental conditions of the surroundings. Rest of the features were removed. So the features that remained were:

- Temperature
- Humidity
- Light Intensity
- Voltage across a 2V battery

B. Data Sampling, Processing

The Intel Berkeley data was used which contained around 2.3 million of data. Since this is a very big chunk of data, it is difficult to use it for our trainng and testing using Machine Learning algorithm process. Therefore sample of the data are taken. 8000 random rows are taken from the data.

Similarly, the sample from the data had also 4 features which are exactly the same as the original data. The original dataset had anomalous values and so did the sampled dataset. The data is then visualized by drawing scatter and box plots of the data. This helps us to visualize the presence of anomalous data.

For analysis Temperature and Light intensity were selected as the features because they gave the highest range in data. As shown by the boxplot, there are visual anomalies. Therefore as a part of data-processing the feature space was reduced from 4 dimensions to 2 dimensions. For works later, only temperature and light intensity was chosen as the features to detect anomalies. 2 shows the box plot of our data to visually represent that the data possesses anomalous values.

C. K-Means Clustering

Since data over here are not labelled to be anomalous or normal, the problem of detecting whether a data instance is anomalous or normal is unsupervised. Therefore, data needs to be clustered. K-means Clustering was done on the data to bring out clusters and separate the normal and anomalous data. The figure 1 shows the scatter plot of our data to visually represent clusters for anomalous and normal data. After clustering, data which belonged to the anomalous cluster (yellow) were labeled as 1. Data which belonged to the normal cluster (purple) were labeled as 0. This gives us the data label and now this data can be used for supervised Machine Learning models.

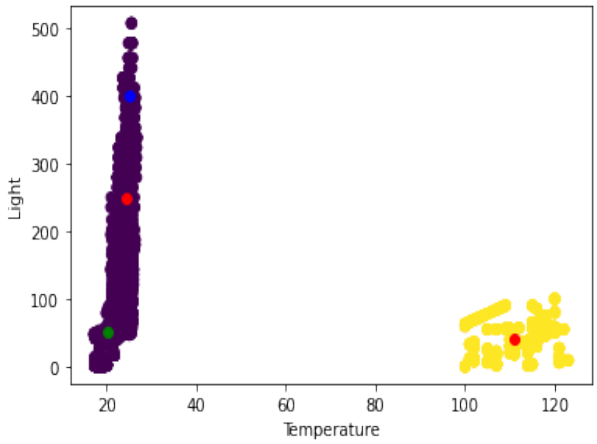

Fig. 1. Scatterplot of Dataset along with clusters

It was also seen selecting K=4 as the number of centroids in K-means clustering gave us very accurate clusters. It is true that the clusters are only two intuitively - anomalous or non-anomalous, but some normal values might lie close to the abnormal cluster's centroid. Therefore, taking 3 centroids for representing the normal cluster of data is a better choice as it can catch the large spread of normal data.

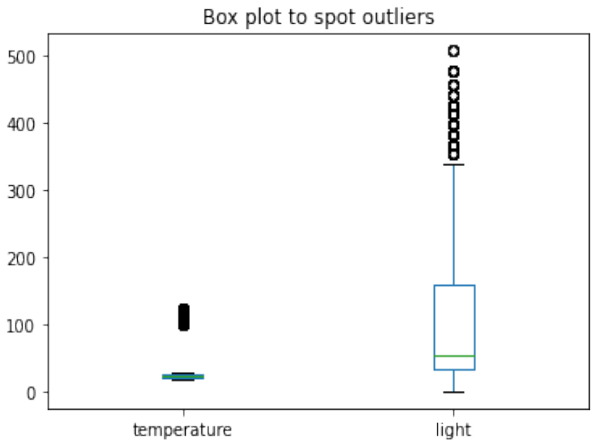

Fig. 2. Box Plot of Dataset

D. Splitting the Data

The sample of data is now labeled. But before proceeding to training of the Machine Learning models with these data, Data needed to be split into training and testing sets. This research used a 80-20 percent split on the data. 80 percent of the data were used to train the models, the rest 20 percent were used to test the models' performances.

The bar chart in figure 3 below depicts the number of samples each set contains.

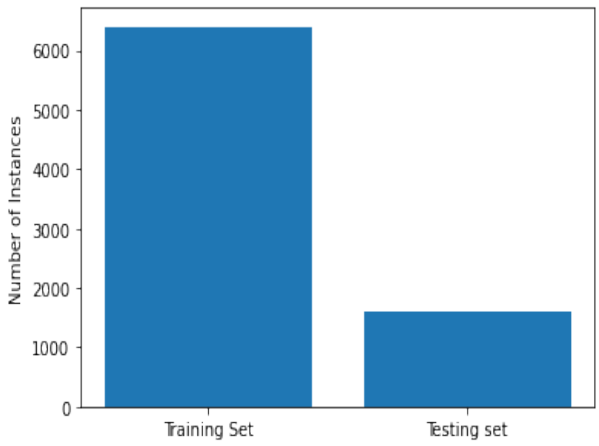

Fig. 3. Bar-chart to represent the number of Training and Testing examples

E. Naive Bayes Classifier

Naive Bayes Classifier is very focused on the Posterior Probability. The Posterior Probability, which is what Naive Bayes is more formally known as, changes the previous belief of an event in light of fresh evidence. The chance of the class

occurring given the new data is the outcome. The Posterior Probability in Navie Bayes Classifier is given by:

$$P(class/features) = \frac{P(class) * P(features/class)}{P(features)} \quad (1)$$

Therefore, it predicts the class of an instance based on its features. The training set was used to fit the Naive Bayes Classifier. Based on the training data, each instance of the testing set's label were classified. The predictions from the Naive Bayes Classifier was then compared to the actual label of the testing set to see how well it performed. No hyper-parameters were adjusted as the default Naive Bayes Classifier gave very good results.

F. Decision Tree Classifier

Decision Tree Classifier splits the data to a tree based on the features. It selects the feature to split the data based on which feature gives the best information gain. If it is a discrete feature, then it splits on the respective possible values of the feature. However if it is a continuous feature, it has many options to split the data. We chose to use the bucket method for splitting the continuous data. The equations to calculate the Entropy, which in turn is used to calculate the Information gain are given below.

$$Entropy(S) = \sum_{i=1}^{c} -p(i)logp(i) \quad (2)$$

$$Gain(S, A) = Entropy(S) - Entropy(A) \quad (3)$$

After the root node is selected based on the feature that gives the highest information gain, the nodes at the secondary level is chosen by who gives the second highest information gain. This process is done until we get pure leaves or near to pure leaves of data (depends on a minimum acceptance threshold) to classify each sample in our test data as anomalous or normal. No hyper-parameters were adjusted as the default Decision Tree Classifier gave very good results.

G. Research Platform

Before the results and analyses are done, it is vital that the platform specifications in which this research is done is shared. Research is done based on a Macbook Pro 16 inch M1 processor with 16GB of RAM.

V. Result and analysis

Different Results were obtained after performing all the ideas described in the Research and Methodology section. However, they need to be discussed. In this section, the discussion of results and analyses on the experiments are done and shared.

A. Value of K in K-Means Clustering

Different Values of K were tried out on K-Means clustering algorithm and the number of misclassified instances were calculated. The number of misclassification was calculated by calculating the instances which belonged to one chunk in the cluster diagram but was assigned to a different cluster. The results are given in Table 1, below.

K	Number of Misclassified Instance
1	3000
2	540
3	150
4	0

TABLE I
Number of Misclassified Instances by varying K in K-Means Clustering

Hence K value was set to 4 for K-Means Clustering as it gave 0 misclassified instances.

B. Classifier Results

The classifier results are attached to the table below. These results are the performance on the testing set. Each classifier was tested for 5 different times. As random sampling was done, the performance varied by a little each time. The metric to represent the performance of the models is Accuracy.

$$Accuracy = \frac{|CorrectlyClassifiedInstances|}{|Dataset|} \quad (4)$$

Run	Accuracy of Decision Tree	Accuracy of Naive Bayes
1	0.960153257	0.933333333
2	0.967816092	0.944061303
3	0.976245211	0.931034483
4	0.973180077	0.936398467
5	0.979124977	0.923697318

TABLE II
Classifiers' Performances

C. Null Hypothesis and Alternate Hypothesis

The Null Hypothesis I am assuming is that both the classifiers perform equally. Therefore, the Alternate Hypothesis becomes that both classifiers are significantly different and one classifier performs better than the other.

Null Hypothesis 1: Both Models perform equally
Alternate Hypothesis 1: Classifiers are dfferent and one is better than the other

A few experiments were done in order to reject the null hypothesis. They are all listed below.

D. Analysis of Variance

Analysis of Variance (ANOVA) is done on the performances of the two models. The test will prove that the models in terms of performance are statistically different. The Anova

test was done on Microsoft Excel. Figure 4 shows the ANOVA experiment. We have found an F value of 63.1757765, which is greater than the F critical value of 5.317655072. This was found at 95% confidence and P value (0.0000457524) is much less than the alpha (0.05). Therefore, I can say with 95 percent confidence that the models' performances are significantly different.

Anova: Single Factor

SUMMARY

Groups	Count	Sum	Average	Variance
Column 1	5	4.85651961	0.97130392	5.6402E-05
Column 2	5	4.6685249	0.93370498	5.5483E-05

ANOVA

ırce of Variat	SS	df	MS	F	P-value	F crit
Between Gro	0.0035342	1	0.0035342	63.1757765	4.5752E-05	5.31765507
Within Grou	0.00044754	8	5.5942E-05			
Total	0.00398174	9				

Fig. 4. Anova Experiment

E. Paired Observations

Paired Observation is done to check whether one algorithm's performance is better than the other. One model can be said to be better than another if the confidence interval at a significant level of the difference in their performance does not include 0.

Since the two models were Decision Tree and Naive Bayes Classifiers, the research included performing paired observations on them. The observations were done at 90, 95 and 99 percent confidence intervals. The intervals of Paired Observation at 90% confidence was (0.032022142, 0.043175742). At 95% confidence, the intervals were (0.030335906, 0.044861977) and at 99% confidence, intervals were (0.02555487, 0.049643013). The results were done extracted by subtracting the Naive Bayes classifier's accuracy from the Decision Tree Classifier's accuracy. Since none of the intervals contained 0, we can say at 90%, 95% and 99% Confidence level that Decision Tree classifier is better than Naive Bayes Classifier. Figure 5 shows the results of Paired Observations test. It is to be noted that all calculations were done on Microsoft Excel.

F. Approximate Visual Test

Approximate Visual Test is yet another test done to check whether one algorithm's performance is better than the other algorithm's performance. In approximate Visual test, the confidence Intervals of both the classifiers' performance is found out. We can only say that a model is better than the other at a given level of confidence only when one model's lower bound is higher than the upper bound, i.e there is no intersecting region in their confidence interval.

Paired Observations			
Trial Number	Result of Decision Tree Classifier (DT)	Result of Naïve Bayes Classifier (	DT-NB
1	0.960153257	0.933333333	0.0268199
2	0.967816092	0.944061303	0.0237548
3	0.976245211	0.931034483	0.0452107
4	0.973180077	0.936398467	0.0367816
5	0.979124977	0.923697318	0.0554277
	Sample Mean		0.0375989
	Variance		0.0001711
	Standard Deviation		0.0130797
	For 90 Percent Confidence Interval, 4 deg freedom		2.1318468
	Lower Bound		0.0320221
	Upper Bound		0.0431757
	For 95 Percent Confidence Interval, 4 deg freedom		2.7764451
	Lower Bound		0.0303359
	Upper Bound		0.044862
	For 99 Percent Confidence Interval, 4 deg freedom		4.6040949
	Lower Bound		0.0255549
	Upper Bound		0.049643

Fig. 5. Paired Observation Experiment

For this research's case, we performed Approximate Visual Test at 90%, 95% and 99% confidence level. The results are all attached to figure 6. It was seen that at all the given confidence levels, Decision Tree Classifier's lower bound was still higher than Naive Bayes Classifier's upper bound. Suggesting that Decision Tree classifier is the better classifier.

Approxiamte Visual Test		
Trial Number	Result of Decision Tree Classifier (DT)	Result of Naïve Bayes Classifier (NB)
1	0.960153257	0.933333333
2	0.967816092	0.944061303
3	0.976245211	0.931034483
4	0.973180077	0.936398467
5	0.979124977	0.923697318
Mean	0.971303923	0.933704981
Variance	5.64019E-05	5.54828E-05
Standard Deviation	0.007510118	0.007448679
90 Percent Confidence Interval	2.131846786	2.131846786
Lower Bound	0.968101838	0.930529092
Upper Bound	0.974506007	0.936880869
95 Percent Confidence Interval	2.776445105	2.776445105
Lower Bound	0.967133637	0.929568811
Upper Bound	0.975474209	0.93784115
99 Percent Confidence Interval	4.604094871	4.604094871
Lower Bound	0.964388463	0.926846096
Upper Bound	0.978219382	0.940563866

Fig. 6. Approximate Visual Test

G. Summary of Results and Hypothesis Testing

The overall summary of all the tests that were done are presented in the table below. The table also points the outcome achieved from each of the tests.

	ANOVA	**Paired Observation Test**	**Approximate Visual Test**
Outcome	Statistically Different	Decision Tree is better	Decision Tree is better

TABLE III
SUMMARY OF RESULTS

Thus, performing these tests, we can say that the models are different and Decision Tree is better as a classifier than Naive

Bayes Classifier. Therefore, we can reject our Null Hypothesis which we assumed initially.

VI. Conclusion and Future Direction

This research paper describes the processing of an environmental IoT dataset and detecting anomalies in the dataset using two different Machine Learning Classifiers. Although both the classifiers have hight accuracy, ANOVA has shown that they are statistically different and Paired Observation and Approximate Visual Test has shown that the Decision Tree classifier is significantly better than the Naive Bayes Classifier.

There are a few steps the research can be extended for the future. Some of them are:

- Trying out different and more complex classifiers like ensemble methods
- Performance of the classifiers on the whole dataset rather than the samples.
- Getting a more recent dataset with more features to represent the environment with more variance within each feature

VII. Acknowledgement

The Intel Lab data was used to perform out classification task. The data can be found from here: http://db.csail.mit.edu/labdata/labdata.html

For Data Processing and Applying Machine Learning algorithms, Scikit Learn and Pandas libraries of Python were used.

References

[1] I. Muhic and M. I. Hodzic, "Internet of things: Current technological review," *PERIODICALS OF ENGINEERING AND NATURAL SCIENCES*, vol. 2, no. 2, pp. 1–8, 2014.

[2] M. Abomhara and G. M. Køien, "Cyber security and the internet of things:vulnerabilities, threats, intrudersand attacks," *Journal of Cyber Security*, vol. 4, pp. 65–88, 2015.

[3] P. Middleton, J. Tully, and P. Kjeldsen, "The internet of things, worldwide, 2013." [Online]. Available: https://www.gartner.com/en/documents/2625419/forecast-the-internet-of-things-worldwide-2013

[4] D. Alexander, A. Finch, D. Sutton, A. Taylor, and A. Taylor, *Information Security Management Principles (2nd Eds.)*. Swindon, United Kingdom: BCS, The Chartered Institute for IT, 2013.

[5] A. Jain, B. Verma, and J. L. Rana, "Anomaly intrusion detection techniques: A brief review," *International Journal of Scientific & Engineering Research*, vol. 5, no. 7, pp. 1372–1383, 2014.

[6] D. K. Prabha and S. S. Sree, "A survey on ips methods and techniques," *International Journal of Computer Science Issues*, vol. 13, no. 2, pp. 38–43, 2016.

[7] V. Timčenko and S. Gajin, "Machine learning based network anomaly detection for iot environments," *ICIST*, vol. 1, pp. 196–201, 2018.

[8] R. Fu, K. Zheng, D. Zhang, and Y. Yang, "An intrusion detection scheme based on anomaly mining in internet of things," pp. 315–320, 2011.

[9] S. S. Tanpure, J. Jagtap, and G. D. Patel, "Intrusion detection system in data mining using hybrid approach," *International Journal of Computer Applications*, pp. 18–21, 2016.

[10] S. Mukherjee and D. Sharma, "Intrusion detection using naive bayes classifier with feature reduction," *Procedia Technology*, vol. 4, pp. 119–128, 2012.

[11] A. Alghuried, "A model for anomalies detection in internet of things (iot) using inverse weight clustering and decision tree," *Semantic Scholar*, 2017.

[12] S. Madden, "Intel lab data." [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html

[13] M. Hasan, M. M. Islam, M. I. I. Zarif, and M.M.A.Hashem, "Attack and anomaly detection in iot sensors in iot sites using machine learning approaches," *Elsevier*, vol. 7, 2019.

[14] Moustafa, Nour, and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," 2015.

[15] N. Moustafa and J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set," *Information Security Journal: A Global Perspective*, vol. 25, no. 1–3, pp. 18–31, 2016.

[16] V. Golmah, "An efficient hybrid intrusion detection system based on c5.0 and svm," *International Journal of Database Theory and Application*, vol. 7, no. 2, pp. 59–70, 2014.