A Statistical Method For Reconnaissance Malware Detection In Internet Of Things (IOT) Network Traffic

by
Devere Anthony Weaver
Advisor: Anthony Okafor, PhD

An Undergraduate Proseminar
In Partial Fulfillment of the Degree of
Bachelor of Science in Mathematics
The University of West Florida
August 3, 2022

The Proseminar of Devere Anthony Weaver is approved:

Dr. Anthony Okafor, PhD, Proseminar Advisor	Date
Dr. Jossy Uvah, PhD, Committee Chair	Date

Accepted for the Department/Division:

Dr. Jia Liu, PhD, Chair	Date

Table of Contents

Abstract

This study sought to determine which network features can be useful predictors in developing a binary classifier that can be used classify internet of things (IoT) network traffic. The classifier was designed to use the statistical features of the network traffic in order to determine whether data packets were normal (benevolent) traffic or contained malicious fingerprinting software. This type of classifier can be useful in identifying network pre-attack indicators. The features for the model were selected by their relevance and correlation with other independent features of the dataset. The logistic regression model was used to create the classifier with seven distinct independent features. The performance of this predictive model is measured using precision. The resulting model had a precision of approximately 90% when predicting instances of normal network traffic using an IoT network dataset.

Chapter 1

Introduction

1.1 Background

The Internet of things (IoT) is a term used to describe physical devices with computational processing power and network connectivity that allows them to transmit and exchange data across networks to other devices. The rise of IoT technologies is currently intense and according to projections for the next 10 years, over 125 billion IoT devices are expected to be connected. The expected investments in IoT technologies are also high with expectations being over $109 billion USD by 2021, with a compound annual growth rate of about 7.3% [15]. IoT devices have found many applications across industries including entertainment, medical, industrial, and even Internet-enabled appliances for home automation. There are endless campaigns of how IoT devices can greatly improve the quality of life for those who own them and improve productivity in businesses that rely on them.

While marketing constantly bombards consumers about the benefits of these devices, there are several causes for concern when it comes to implementing IoT devices across both private networks and the public internet. Some of these are related to security, privacy, regulatory, and interoperability issues. This study concerns itself with the issue of security as IoT devices are frequent targets of cyber attacks.

As more businesses and consumers depend on them, IoT device numbers have grown at an unprecedented rate. The billions of devices online means there is an increased attack surface that grows everyday. This large attack surface gives malicious actors numerous and diverse opportunities to carry out their attacks across networks. In addition, IoT devices are targeted because they often lack the processing power to implement known security measures. The lack of processing power prevents basic security measures such as encryption or multiple security layers. Another weakness is that these devices are not always updated by the consumer either due to lack of awareness or by manufacturer design. When this occurs, devices are unable to receive critical security updates.

The first step in a cyber attack is known as reconnaissance. In this phase, attackers actively probe a network and its network-connected devices, seeking information on the layout of the network and any possible vulnerabilities. The most popular method of detection for malicious network traffic is the intrusion detection system (IDS). These systems typically come in two variants, the first being a signature-based IDS and the second an anomaly-based IDS. A signature-based IDS attempts to determine if network data packets are malicious by performing packet inspection where the transmitted data is analyzed and compared against a library of known cyber attacks. An anomaly-based IDS attempts to catch malicious actions across a network

via monitoring for surprising behavior, such as increased network traffic or unauthorized devices connecting to a network.

1.2 Statement of Problem

In order to improve the cybersecurity of IoT networks, a statistical-based intrusion detection system can be implemented. The goal of this study is to build a binary classifier using the logistic regression model. The binary classifier will classify observations as either normal instances of network traffic or malicious instances of reconnaissance traffic.

1.3 Relevance of Problem

A statistical-based IDS is one that attempts to determine whether or not malicious traffic is present in a network by using only the statistical features of the network traffic. This prevents having to do deep packet inspection, as is the case with a signature-based IDS. A statistical-based IDS also allows for the possible prediction of unknown pre-attack indicators. Traditional IDS do not allow for this and an attack must have previously been seen for it to be detected while the statistical-based IDS should ideally be able to generalize what it has learned to predict new attacks. Being able to predict and classify traffic that has reconnaissance software is the first step in preventing a potential full-scale cyber attack.

1.4 Literature Review

Previous literature in the field has been concerned with using statistical and machine learning algorithms to detect malicious software in computers systems and networks. These techniques range from using simple methods such as logistic regression to more complicated computational methods such as convolutional neural networks (CNN). However, most of the research investigating the use of statistical and machine learning algorithms for malware detection revolves around malware detection in traditional networks and computing systems. The available literature for the detection of malware specifically in IoT networks is substantially smaller.

S.J. Rao utilized logistic regression classifiers in an attempt to locate malware within secured systems [14]. The classifiers proposed were able to recognize malware with a likelihood of 74-83%.

In 2011, Ozkan et al. utilized a convolutional neural network in order to analyze and classify different families of malware [11]. The research team was able to achieve 85% accuracy when applying their model to thirty-six malware families across 12,279 samples. Further, they were able to achieve a 99% accuracy when

used on twenty-five malware families across 9,339 malware samples.

In work by Soe et al., the researchers proposed a novel feature selection algorithm that used correlation-based feature selection to calculate the final analysis feature set using the BoT-IoT dataset [15]. The goal was to create a lightweight detection system for cyber attacks in IoT environments using the features present in the BoT-IoT dataset. Their machine learning-based intrusion detection system was implemented on a Raspberry Pi computer. They then used these feature to build statistical models using multiple algorithms.

Win et al. experimented with security analysis on big data. The data was collected from network logs of virtual machines [16]. The researchers extracted what they believed to be the most relevant features using a map-reduce parser. They then went on to use logistic regression to calculate the probability of attacks in the virtual environment. They achieved an accuracy of 98.84%.

Research performed by Dermipolat et al. proposed a new framework for IoT security named ProtEdge [2]. Their proposed system is a Software-defined network (SDN) architecture that utilizes few-shot learning in an attempt to accurately detect IoT network intrusions. The model was evaluated using three IoT network datasets: Bot-IoT, UNSW-NB15, and a custom SDN data set.

Bapat et al. also used logistic regression in an experiment to create a statistical-based intrusion detection system [1]. They were specifically testing for botnet network traffic. They achieved an accuracy of 95% and recall of 96.7%.

Chapter 2

Data and Methods

2.1 Data Description

The data used for this study comes from the BoT-IoT dataset developed by a group of researchers at the Intelligent Security group of the University of New South Wales (UNSW) in Canberra, Australia. The technical details of the dataset are documented in [3-9]. It is a labelled dataset that was developed using multiple virtual machines with various operating systems, firewalls, network taps and two software security tools [12].

The dataset is available as four different sets. The first set is the raw dataset that contains 69.3GB of raw packet capture files that must be preprocessed using networking tools before being useful for any analysis. The second set, referred to as the Full Set, is composed of comma separated value files only. There are 73 million instances in this subset with twenty-six independent and three dependent features. The second subset, referred to as the 5% Subset, contains 3.6 million instances with forty-three independent features and three dependent features. This set includes all of the features from the Full Set and includes variables that were derived from them. The final subset is referred to as the 10-Best Subset. This set contains the same 3.6 million instances as the 5% Subset but it only has ten features. These features were derived through the mapping of the correlation coefficient and joint entropy of the forty-three independent features [12].

For this research project, the 5% Subset was used as it contains all forty-three independent features but less observations that need to be processed. In order to determine which features are the most useful for creating a predictive model, specific features may need to be removed before analysis.

The first set of features to be dropped from analysis were established in previous literature as invalid features. In [12], Peterson et al. analyzed each feature and determined that six of the independent features were irrelevant due to these features being highly specific to the testing environment in which the data were generated. Environment specific data has a tendency to cause overfitting in a model which can reduce the model's generalizability. There are also three variables (State, Protocol, Flags) that can be considered as duplicates. These three features are categorical but there also exists encoded features that correspond to them. Therefore, they don't contain any new information and were dropped from consideration. This leaves thirty-four independent features remaining to consider for analysis.

The remaining independent features were tested for correlation using Pearson's Correlation Coefficient. Using this statistic, the correlation between two independent features is computed to determine the strength and direction of their relationship. Numerically, Person's Correlation Coefficient is defined as the ratio

of the covariance of two independent variables divided by the product of the standard deviations of each independent variable.

Those features where the absolute value of their computed Pearson's Correlation Coefficient is greater than 0.7 were considered highly correlated thus they were dropped for consideration to be used in the analysis. To easily visualize the correlation coefficient for all independent features, a correlation matrix is often used. Table 2.1 below contains a snippet of the correlation matrix of the data for brevity. It includes 5 of the features that were dropped for being highly correlated with other features and one feature that was used for analysis.

	packets	bytes	sum	min	max	flow state flags
packets	1.000					
bytes	0.984	1.000				
sum	0.833	0.750	1.000			
min	0.707	0.639	0.791	1.000		
max	0.748	0.671	0.912	0.795	1.000	
flow state flags	0.044	0.051	0.006	−0.015	−0.015	1.000

Table 2.1: Snippet of correlation matrix.

The project is concerned with developing a binary classifier to use as a predictive model. Correlated features typically do not have much effect on the outcome of a prediction, but the project is also concerned with determining which features are considered important for making those predictions. In an effort to isolate which of these network features are important to the predictive model, highly correlated features should be removed. After testing for correlation, twenty features were considered highly correlated and were removed from the analysis. Of the remaining fourteen independent features, seven were removed from consideration. After further analysis of their descriptions, it was determined that these features are aggregates that utilize information from other features. For this study's goals, it is undesirable to have multiple features in the model that measure the same phenomena. This leaves seven independent features remaining for analysis. Table 2.2 below contains the independent feature names used for analysis and the dichotomous independent variable.

Selected Features	
Feature Name	Type
Flow state flags	Categorical (encoded, 6 levels)
Network protocol	Categorical (encoded, 3 levels)
Destination port number	Numeric (0, 65534)
Duration transmission	Numeric (0, 1940.85)
Source port number	Numeric (0, 65534)
Total packets per second in transaction	Numeric (0, 4.96)
Destination-to-source packets per second	Numeric (0, 4037.5)
Attack (dependent)	Categorical (0 = Normal, 1 = Reconnaissance)

Table 2.2: Uncorrelated features used for analysis, their data types, and the dependent variable.

The BoT-IoT dataset contains three possible dependent features: Attack, Category, and Subcategory. Each of these can be used for a different type of analysis but for this study the dependent feature Attack was used. This choice was made because a binary classifier will be the end result. The Attack dependent feature is a binary categorical variable with a value of 0 being normal network traffic and a value of 1 being malicious reconnaissance traffic while Category and Subcategory contain more than two classes each.

The dataset contains millions of observations; therefore, it needed to be downsampled to a reasonable size. The final sample size for analysis was 5,000 observations. This sample includes all observations with an Attack category of Normal and observations with an Attack category of Reconnaissance. The sample then contained 430 instances of Normal traffic and 4570 instances of Reconnaissance traffic.

The final step before analysis is to split the sample. The logic is to create two smaller, distinct sets where one can be used to train the model and the second can be used to test the performance of the created model. The typical training-testing split for model development is 80% of our sample goes in the training set and the remaining 20% in the testing set. This results in our training set containing 4,000 observations and the testing set containing 1,000.

2.2 Models and Assumptions

The logistic regression model was used for analysis. Described by Klienbaum et al., "Logistic regression is a statistical modeling approach that can be used to describe the relationship of several predictor variables to a dichotomous dependent variable" [3]. This model is useful for applications where the independent feature can take on one of two values. The logistic regression model is given by

$$P(Y = 1) = \frac{1}{1 + exp\left[-(\beta_0 + \sum_{j=1}^{k} \beta_j X_j)\right]}, \tag{2.1}$$

where β_0 is the coefficient of the intercept term, β_j is the coefficient corresponding to the j^{th} observation, and X_j is the observed value of the j^{th} observation. This model above describes the probability of occurrence of one of the two possible outcomes of our event Y. In the context of this study, event Y is the outcome that our observation is either classified as normal traffic or is classified as malicious traffic.

The assumptions that should be met before the attempting to implement the logistic regression model for analysis are:

1. A dichotomous response variable
2. Independent observations
3. No or minimal multicollinearity between independent variables
4. Large sample size
5. Linear relationship between the independent variables and the logit form of the response variable

Typically, the first step in statistical modeling is to create a mathematical model that describes the phenomena. For the logistic regression model, the corresponding mathematical function is the logistic function,

$$f(z) = \frac{1}{1 + e^{-z}} \tag{2.2}$$

where

$$z = \beta_0 + \sum_{j=1}^{k} \beta_j X_j,$$

and β_0 is the coefficient of the intercept term, β_j is the coefficient corresponding to the j^{th} observation, and X_j is the observed value of the j^{th} observation.

The logistic function is useful for modeling the probability of dichotomous response variables since its domain is the set of all real number and its range is the interval $[0, 1]$. Probability values can only take on a

value in the interval $[0, 1]$.

After selecting the proper mathematical model of the phenomena, the model must be fit to the data used for analysis. For logistic regression, the method of maximum likelihood is used to fit the model to the data and estimate the unknown coefficients. While the mechanics of the method of maximum likelihood are outside the scope of this research project, the algorithm is implemented by statistical analysis software and was utilized to fit the model to the training dataset used for analysis.

Once the model is fit to the data, interpretation of the model is important for statistical inference. The coefficients for a logistic regression model are often used to estimate a parameter called the odds ratio (OR). The odds ratio is a parameter that can be used to measure the effect of independent features on the dependent feature. The odds ratio is defined as the ratio of the probability of an event occurring divided by the probability of that same event not occurring. The odds ratio is given by,

$$Odds(Y) = \frac{P(Y)}{1 - P(Y)} \tag{2.3}$$

where $P(Y)$ is the probability of some event Y occurring and $1 - P(Y)$ is the probability of event Y not occurring.

In order to compute an odds ratios, the logistic regression model must be transformed to what is known as the logit form of the model. The logit transformation of the model is defined as the natural log of the odds of the event Y. Taking the natural log of the logistic regression model gives an expression that is a linear relationship between the independent variables, their coefficients, and the dichotomous dependent variable. The logit form is given by,

$$ln\left[\frac{P(Y=1)}{1 - P(Y=1)}\right] = \beta_0 + \sum_{j=1}^{k} \beta_j X_j, \tag{2.4}$$

where β_0 is the coefficient of the intercept term, β_j is the coefficient corresponding to the j^{th} observation, and X_j is the observed value of the j^{th} observation.

The importance of computing the odds ratio for a given event lies in model interpretation. As previously mentioned the odds ratio is used to measure the effect of independent variables on the probability of the dependent variable. This practical interpretation for a given odds ratio is "for a 1 (unit) increase in (independent feature), we expect the odds of (outcome) to be multiplied by (odds ratio for coefficient)." For example, if an odds ratio for a given coefficient was computed to be 2.5, then for a 1 unit increase in the independent feature, we expect the odds of our outcome to be 2.5 times higher. Note, an odds ratio of 1 would indicate that there is no effect of an independent variable on the outcome of the dependent variable.

2.3 Analysis and Results

In order to practically fit a model to data, software is used which implements the most common and useful estimation algorithms. For this analysis, the statistical computing software used was R. R is a common open source statistical computing package [13]. Using R, the model was fit to the training data containing the 4,000 observations across 7 features. The logistic regression model coefficient estimates and their standard errors are summarized in the Table 2.3.

Model Results		
Feature Name	Coefficient	Standard Error
Intercept	4.62	6.25×10^{-1}
Flow state flags	1.73	3.82×10^{-1}
Network protocol	-2.90	1.64×10^{-1}
Destination port number	2.80×10^{-4}	3.07×10^{-5}
Duration transmission	-0.014	1.81×10^{-3}
Source port number	-1.32×10^{-6}	7.00×10^{-6}
Total packets per second in transaction	9.70×10^{-4}	4.07×10^{-4}
Destination-to-source packets per second	-1.51×10^{-4}	1.29×10^{-3}

Table 2.3: Summary of fitted model results.

These estimates mean the logit transformation of the logistic regression model is given by,

$$ln\left[\frac{\hat{\pi}}{1-\hat{\pi}}\right] = 4.62 + 1.73 flow\_state\_flags - 2.90 network\_protocol + 2.80 \times 10^{-4} destination\_port\_number$$

$$-0.014 total\_duration\_transmission - 1.32 \times 10^{-6} source\_port\_number + 9.70 \times 10^{-4} total\_packets\_per\_second$$

$$-1.51 \times 10^{-4} destination\_to\_source.$$

As previously mentioned, the odds ratios are often computed for the purposes of statistical inference. While this is important for explanatory purposes, for the purposes of prediction they are not as relevant.

The end goal of this analysis is prediction, and as such does not concern itself with measuring the numerical effect of individual variables on the dependent variable. Instead, it is concerned only with which variables are useful for making accurate predictions on unseen data.

Now that the model has been built, its predictive performance and its ability to generalize must be tested. To do so, the testing dataset must be used with the model to create a prediction for each observation in the testing set. These predicted labels must be compared with the actual labeled value of the observation. Again, using R, the model was tested using the testing data set. To easily visualize the performance of the model, a confusion matrix is presented. A confusion matrix is a type of a contingency table where the rows of the table are the actual true labels for each class and the columns of the table are the predicted values for each class.

		Predicted		
		Normal	Reconnaissance	Total
Actual	Normal	86	0	86
	Reconnaissance	9	905	914
	Total	95	905	1,000

Table 2.4: Confusion matrix of testing results.

In a confusion matrix, we have four important values. The value in the top left represents what is known as a true positive. True positives are outcomes where the model correctly predicted the positive class. In this case, the model was able to correctly predict 86 instances of the Normal class found in the testing data set. Below this cell is what is known as the false positive. The false positives are the instances where the model incorrectly labeled the observation as belonging to the Normal class when it actually belongs to the Reconnaissance class. In the second column, the first cell represents the number of false negatives. Here, the false negatives are instances where the model labeled Normal traffic as belonging to the Reconnaissance class. Finally, below that is the cell containing the true negatives. True negatives are the instances where the model correctly labeled malicious traffic as belonging to the Reconnaissance class.

There exist many different performance metrics for evaluating predictive models. The most common metric is known as accuracy. Accuracy is simply a measure of how often the model was able to correctly label the observations. In order to compute the accuracy, the following is used

$$ACC = \frac{TP + TN}{Total}. \tag{2.5}$$

This equation tells us the accuracy is simply the ratio of the sum of the true positives and true negatives divided by the total number of observations. For the model, the accuracy is then 0.991. Typically, the

accuracy is given as a percentage, thus the accuracy is 99.1%. While an accuracy of 99.1% can be seen as a high-performing model, this metric may not necessarily be the best one depending on the situation.

Recall, the model is attempting to successfully predict which instances of malicious reconnaissance traffic in an IoT network. For this type of classifier, one should be most concerned with correctly labelling malicious traffic. If instances of Normal traffic are labeled as malicious, then this is simply a false alarm and no real harm to the network is done. However, if malicious traffic is able to go undetected by being mislabeled, then there can be harmful impact done to the network. This being the case, it can be argued the better performance metric would be to use what is known as precision.

Precision is a measure of how often a model was correct in predicting positive labels. In other words, precision measures how often the model was able to correctly label instances of normal traffic as Normal traffic. Ideally, precision should be maximized in this scenario as it means the model is infrequently allowing malicious traffic to go by undetected. To compute the measure of precision, we use

$$PPV = \frac{TP}{TP + FP}. \tag{2.6}$$

The abbreviation PPV is short for positive predicted value and it is equal to the number of true positives divided by the sum of the true positive and false positives. Overall, the model has a precision of 0.905, or 90.5% as a percentage. Practically, this means roughly 10% of the time, the model was allowing malicious traffic to go undetected. Using the testing set, it also never mislabeled instances of the Normal class as belonging to the Reconnaissance class.

Chapter 3

Conclusions

3.1 Summary of Key Findings

Using the BoT-IoT dataset, it was determined there were seven uncorrelated independent features that appear to be useful for classifying network traffic as malicious reconnaissance traffic. With a precision of approximately 90%, the model performed reasonably well at its task on the testing dataset. Using the logistic regression model to build a binary classifier with meaningful features can result in a predictive model that is able to detect and a classify malicious traffic from normal IoT network traffic. There is still room for improvement in order to maximize precision; however, the results from this model work as a good baseline performance that can be later improved.

3.2 Suggestions for Further Study

One limitation when working with the BoT-IoT dataset is that the dataset is unnaturally imbalanced. Suggestions for future study include possibly implementing methods to balance the training dataset, for example, random oversampling or random undersampling of the minority class (normal traffic). Random oversampling or random undersampling can be performed on the training dataset before analysis in an attempt to make even the number of observations per class.

Another suggestion for future study is to train a predictive model with a completely different algorithm while using the same BoT-IoT dataset. There exist numerous algorithms that can be used to model the outcome of a dichotomous dependent variable. Some examples include Decision Trees, Random Forests, and Support Vector Machines to name a few. Using different algorithms can result in different models which may or may not have improved performance. These new models can be compared with the one created used the logistic regression model.

Bibliography

[1] Bapat, R., Mandya, A., Liu, X., Abraham, B., Brown, D. E., Kang, H., Veeraraghavan, M. (2018). Identifying malicious botnet traffic using logistic regression. 2018 Systems and Information Engineering Design Symposium (SIEDS). https://doi.org/10.1109/sieds.2018.8374749

[2] Demirpolat, A., Sarica, A. K., Angin, P. (2020). ProtÉdge: A few-shot ensemble learning approach to software-defined networking-assisted edge security. Transactions on Emerging Telecommunications Technologies, 32(6). https://doi.org/10.1002/ett.4138

[3] Kleinbaum, D. G., Kupper, L. L., Nizam, A., Rosenberg, E. S. (2013). Applied Regression Analysis and Other Multivariable Methods (5th ed.). Cengage Learning.

[4] Koroniotis, Nickolaos, Nour Moustafa, Elena Sitnikova, and Benjamin Turnbull. "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset." Future Generation Computer Systems 100 (2019): 779-796.

[5] Koroniotis, Nickolaos, Nour Moustafa, Elena Sitnikova, and Jill Slay. "Towards developing network forensic mechanism for botnet activities in the iot based on machine learning techniques." In International Conference on Mobile Networks and Management, pp. 30-44. Springer, Cham, 2017.

[6] Koroniotis, Nickolaos, Nour Moustafa, and Elena Sitnikova. "A new network forensic framework based on deep learning for Internet of Things networks: A particle deep framework." Future Generation Computer Systems 110 (2020): 91-106.

[7] Koroniotis, Nickolaos, and Nour Moustafa. "Enhancing network forensics with particle swarm and deep learning: The particle deep framework." arXiv preprint arXiv:2005.00722 (2020).

[8] Koroniotis, Nickolaos, Nour Moustafa, Francesco Schiliro, Praveen Gauravaram, and Helge Janicke. "A Holistic Review of Cybersecurity and Reliability Perspectives in Smart Airports." IEEE Access (2020).

[9] Koroniotis, Nickolaos. "Designing an effective network forensic framework for the investigation of botnets in the Internet of Things." PhD diss., The University of New South Wales Australia, 2020.

[10] Nižetić, S., Šolić, P., López-de-Ipiña González-de-Artaza, D., Patrono, L. (2020). Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. Journal of Cleaner Production, 274, 122877. https://doi.org/10.1016/j.jclepro.2020.122877

[11] Ozkan, K., Isik, S., Kartal, Y. (2018). Evaluation of convolutional neural network features for malware detection. 6th International Symposium on Digital Forensic and Security (ISDFS), 1–5.

[12] Peterson, J. M., Leevy, J. L., Khoshgoftaar, T. M. (2021). A Review and Analysis of the Bot-IoT Dataset. IEEE International Conference on Service-Oriented Systems Engineering (SOSE), 20–27.

[13] R: The R Project for Statistical Computing. (n.d.). The R Project for Statistical Computing. Retrieved July 1, 2022, from https://www.r-project.org

[14] Rao, S. J. (2003). Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis. Journal of the American Statistical Association, 98(461), 257–258. https://doi.org/10.1198/jasa.2003.s263

[15] Soe, Y. N., Feng, Y., Santosa, P. I., Hartanto, R., Sakurai, K. (2020). Towards a Lightweight Detection System for Cyber Attacks in the IoT Environment Using Corresponding Features. Electronics, 9(1), 144. https://doi.org/10.3390/electronics9010144

[16] Win, T. Y., Tianfield, H., Mair, Q. (2018). Big Data Based Security Analytics for Protecting Virtualized Infrastructures in Cloud Computing. IEEE Transactions on Big Data, 4(1), 11–25. https://doi.org/10.1109/tbdata.2017.271533