**Use of Machine Learning in Detecting Network Security of Edge Computing System**

**-Abhinav Kohli**

**8005117285**

**Abstract**

The Internet of Things (IoT) can increase efficiency, but it also comes with new risks. Have you started a sentence today with “Alexa,” “OK Google,” or “Siri”? If so, you may have put your home or business at risk. The Internet of Things (IoT) is exploding, but many people are still unaware of the risks introduced by smart devices. With the increased use of IoT infrastructure in every domain, threats and attacks in these infrastructures are also growing commensurately.Edge computing is transforming the way data is being handled, processed, and delivered from millions of devices around the world. The explosive growth of internet-connected devices – the IoT – along with new applications that require real-time computing power, continues to drive edge-computing systems. IoT devices provide us with a broad new set of capabilities, but they also introduce many potential security vulnerabilities. Every smart device needs to be protected and maintained over time as new vulnerabilities are detected.

Faster networking technologies, such as 5G wireless, are allowing for edge computing systems to accelerate the creation or support of real-time applications, such as video processing and analytics, self-driving cars, artificial intelligence and robotics, to name a few.

**Introduction**

Smart device manufacturers are putting IoT sensors into cars, light bulbs, electric outlets, refrigerators, and many other appliances. These sensors connect the devices to the internet and allow them to communicate with other computer systems. It’s estimated that by 2022, there will be 50 billion consumer IoT devices worldwide.

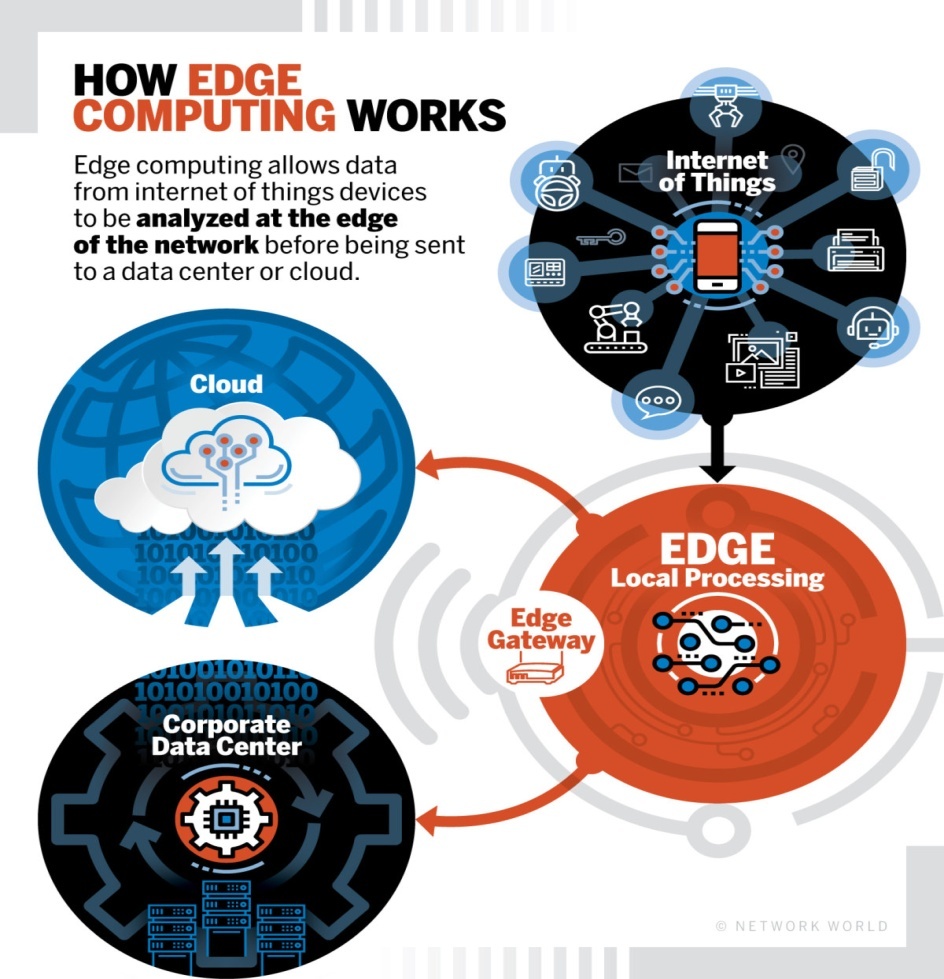
IoT devices provide us with a broad new set of capabilities, but they also introduce many potential security vulnerabilities. Every smart device needs to be protected and maintained over time as new vulnerabilities are detected. That leads us to the second problem. Many IoT devices don’t include security features. Perhaps the vendor was in too much of a hurry to get its new smart device into the market, or perhaps it was too challenging to develop a security interface. While the devices themselves have become more sophisticated, there’s often no underlying IoT security framework to protect them. IoT devices use a wireless medium to broadcast data which makes them an easier target for an attack.. Normal communication attack in the local network is limited to local nodes or small local domain, but attack in IoT system expands over a larger area and has devastating effects on IoT sites.

Thenceforth, a secured IoT infrastructure is necessary for the protection from cybercrimes. The security measures that have been used become vulnerable with the vulnerability of IoT devices.

**Edge computing**

At its basic level, edge computing brings computation and data storage closer to the devices where it’s being gathered, rather than relying on a central location that can be thousands of miles away. This is done so that data, especially real-time data, does not suffer latency issues that can affect an application’s performance. In addition, companies can save money by having the processing done locally, reducing the amount of data that needs to be processed in a centralized or cloud-based location.

Edge computing was developed due to the exponential growth of IoT devices, which connect to the internet for either receiving information from the cloud or delivering data back to the cloud. And many IoT devices generate enormous amounts of data during the course of their operations.



Since the data doesn’t have to travel all the way back to the central server for the device to know that a function needs to be executed, edge computing networks can greatly reduce latency and enhance performance. The speed and flexibility afforded by this approach to handling data creates an exciting range of possibilities for organizations.

**The 5 Best Benefits of Edge Computing**

**• Speed**

**• Security**

**• Scalability**

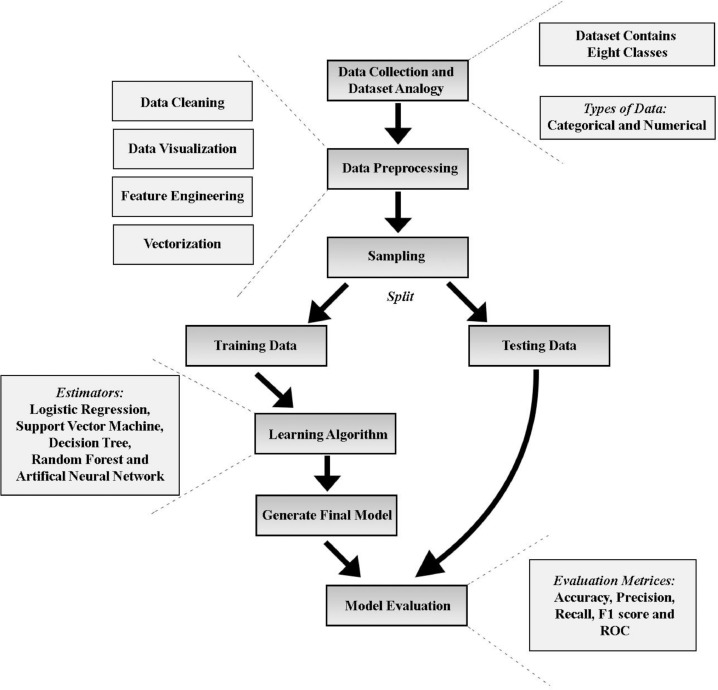
**• Versatility**

**• Reliability**

Devices connected to the internet generate huge amounts of data that provides an enormous opportunity to businesses, but also an equally enormous challenge in terms of managing, analyzing, and storing that data. Traditionally, these processes were handled in a company’s private cloud or data center, but the sheer volume of data has strained these networks to their absolute limits.

Edge systems alleviate this pressure by pushing data processing away from a centralized core and distributing it among local edge data centers and other devices closer to the source. Analyzing data closer to where it’s collected provides huge benefits in terms of cost and efficiency. By utilizing edge systems, companies can also address problems associated with low connectivity and the cost of transferring data to a centralized server.

**Material and method**



**Finding the model for detecting attacks on system**

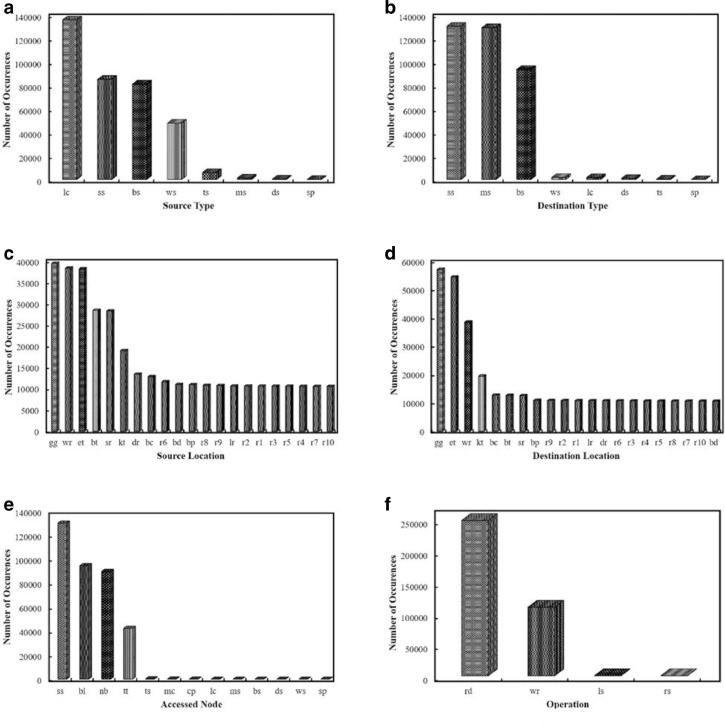
The first process of this framework is the dataset collection and dataset observation. In this process, the dataset was collected and observed meticulously to find out the types of data. Besides, data preprocessing was implemented on the dataset. Data preprocessing consists of cleaning of data, visualization of data, feature engineering and vectorization steps. These steps converted the data into feature vectors. These feature vectors were then split into 80–20 ratio into training and testing set. The training set was used in Learning Algorithm, and a final model was developed using an optimization technique. Different classifiers used in this work employed different optimization techniques. Logistic Regression used coordinate descent. SVM and ANN used conventional gradient descent technique. The optimizer is not used in the case of DT and RF because these are non-parametric models. The final model was evaluated against the testing set using different evaluation metrics.

**Data collection**

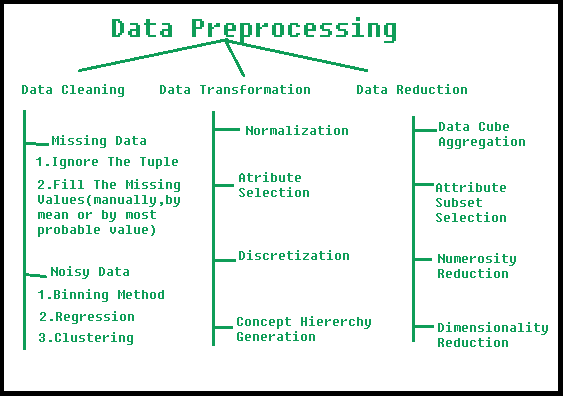
The open source dataset was collected from kaggle provided by Pahl et al.. They have created a virtual IoT environment using Distributed Smart Space Orchestration System (DS2OS) for producing synthetic data. Their architecture is a collection of micro-services which communicates with each other using the Message Queuing Telemetry Transport (MQTT) protocol. In the dataset, there are 357,952 samples and 13 features. The dataset has 347,935 Normal data and 10,017 anomalous data and contains eight classes which were classified. Features “Accessed Node Type” and “Value” have 148 and 2050 missing data, respectively.

**Frequency distribution of considered attacks**

| Attacks | Frequency | % of Total | % of Anomalous |
| --- | --- | --- | --- |
|  | Count | Data | Data |
| Denial of Service | 5780 | 01.61% | 57.70% |
| Data Type Probing | 342 | 00.09% | 03.41% |
| Malicious Control | 889 | 00.24% | 08.87% |
| Malicious Operation | 805 | 00.22% | 08.03% |
| Scan | 1547 | 00.43% | 15.44% |
| Spying | 532 | 00.14% | 05.31% |
| Wrong Setup | 122 | 00.03% | 01.21% |
|  |  |  |  |



### Data preprocessing



## Logistic Regression (LR)

Logistic Regression (LR) is a discriminative model which depends on the quality of the dataset. Given the features X=X1,X2,X3,…,Xn (where,X1−Xn=Distinctfeatures), weights W=W1,W2,W3,…,Wn, bias b=b1,b2,…,bn and Classes C=c1,c2,…,cn (in our case, we have eight classes) the equation for estimation of posterior is given in following (1) PredictedValue:p(y=C|X;W,b)=11+exp(−WtransposeX−b)

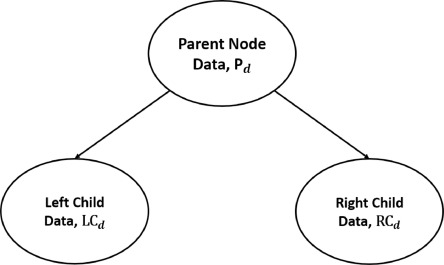
## Support Vector Machine (SVM)

Support Vector Machine is another discriminative model like LR. It is a supervised learning model for analyzing the data used for classification, regression, and outliers detection. SVM is most applicable in the case of Non-Linear data. Given Input x, Class or Label c and LaGrange multipliers α; weight vector Θ can be calculated by following equation:

(2)Θ=∑i=1mαicixiThe target of the SVM is to optimize the following equation:(3)Maximizeαi∑i=1mαi−∑i=1m∑j=1mαiαjcicj<xixj> < xi, xj >  is a vector which can be obtained by different kernels like polynomial kernel, Radial Basis Function kernel and Sigmoid Kernel.

## Decision Tree (DT)

Decision Tree allows each node to weigh possible actions against one another based on their benefits, costs, and probabilities. Overall, it is a map of the possible outcomes of a series of related choices. A DT generally starts with a single node and then it branches into possible outcomes. Each of these outcomes lead to additional nodes, which branch off into other instances. So from there, it became a tree-like shape; in other words, a flowchart-like structure. Considering a binary tree where a parent node is split into two children node a left child and a right child. Parent node, left child and right child contains data Pd, LCd, RCd, respectively. Given, features x, impurity measure I(data), the number of samples in parent node Pn, the number of samples in left child LCn and the number of samples in right child RCn;DT’s target is to maximize following Information Gain in InformationGain(Pd,x)=I(Pd)−LCnPnI(LCd)−RCnPnI(RCd)Impurity Measure I(data) can be calculated in three techniques Gini Index IG, Entropy IH and Classification Error IE.



**Decision Tree Splitting.**

## Random Forest (RF)

As the name implies, the random forest algorithm creates the forest with many decision trees. It is a supervised classification algorithm. It is an attractive classifier due to the high execution speed. Many decision trees ensemble together to form a random forest, and it predicts by averaging the predictions of each component tree. It usually has much better predictive accuracy than a single decision tree. In general, the more trees in the forest the more robust the forest looks.

**Artificial Neural Network (ANN)**

Artificial Neural Network (ANN) is a machine learning technique which is the skeleton of different deep learning algorithms. We can train the ANN model using raw data. Compared to other classifiers it has a large number of parameters for tuning which makes it a complex structure. It also takes a long time to optimize error than other techniques. For this reason, Neural Network algorithm instances are trained in Graphics Processing Unit using CUDA programming. Each single Neuron Node of ANN is trained with feature set X=X1,X2,X3,…,Xn (where,X1−Xn=Distinctfeatures). The features are multiplied by some random weights, W=W1,W2,W3,…,Wn and added with bias values, b=b1,b2,…,bn. The values are then given as input in non-linear activation function. Activation functions can be of several types. Following are some activation functions. In the equations, (i) means a single sample.(8)SigmoidFunction:σ(z)ora(z)=11+e−z(9)TanhFunction:a(z)=ez−e−zez+e−z(10)RectifiedLinearUnit(RELU):a(z)=max(0,z)(11)LeakyRELU:a(z)=max(0.001\*z,z)After applying Non-Linear function, a softmax function is applied to get initial predicted value which is shown in PredictedValue:y^(i)=σ(WtransposeX(i)+b)Lastly from the true value and the predicted value, the loss function is calculated and weights of the whole neural network architecture is modified using the backpropagation technique, gradient descent and error got from the loss function. The equation of loss function is given in the following equation:(13)L(y^(i),y(i))=−(y(i)log(y^(i))+(1−y(i))log(1−y^(i)))

### Evaluation criteria

The following metrics were calculated for evaluating the performance of the developed system. Using these metrics, one can decide which technique is best suited for this work.

## Confusion matrix

The confusion matrix is used to visualize the performance of a technique. It is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It allows for easy identification of confusion between classes. Most of the time, almost all performance measures are computed from it. A confusion matrix is a summary of prediction results on a classification problem. A definition of True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN) for multiple classes can be given from confusion matrix. Let Ci be any class out of the eight classes. Following are the definitions of TP, FP, FN, and TN for Ci:

* TP(Ci) = All the instances of Ci that are classified as Ci.
* FP(Ci) = All the non Ci instances that are classified as Ci.
* FN(Ci) = All the Ci instances that are not classified as Ci.
* TN(Ci) = All the non Ci instances that are not classified as Ci.

## Accuracy

A model’s accuracy is only a subset of the model’s performance. Accuracy is one of the metrics for evaluating classification models. depicts single class accuracy measurement.

**Accuracy=TruePositive+TrueNegativeTruePositive+TrueNegative+FalsePositive+FalseNegative**

## Precision

Precision means the positive predictive value. It is a measure of the number of true positives the model claims compared to the number of positives it claims. The precision value for a single class is given in the following equation:

**Precision=TruePositiveTruePositive+FalsePositive**

## Recall

The recall is known as the actual positive rate which means the number of positives in the model claims compared to the actual number of positives there are throughout the data. The recall value for a single class is given in the following equation:

**Recall=TruePositiveTruePositive+FalseNegative**

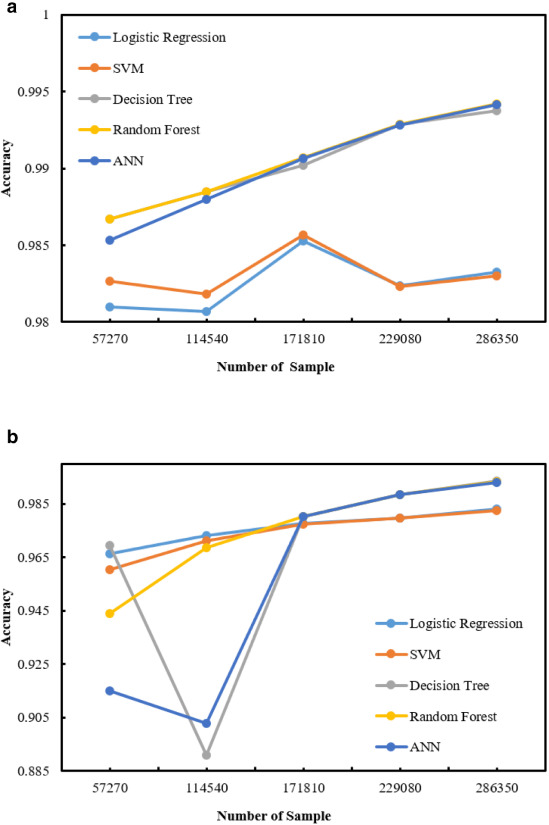
## F1 score

The F1 score can also measure a model’s performance. It is a weighted average of the precision and recall of a model. The F1 Score value for a single class is given in equation :

**F1Score=2\*TruePositive2\*TruePositive+FalsePositive+FalseNegative**

## Result analysis

In the Data Analysis subsection, it has been described that several machine learning techniques were applied to the dataset. Five-fold cross-validation was performed on the dataset using each of these techniques. [Fig. 4](https://www.sciencedirect.com/science/article/pii/S2542660519300241" \l "fig0004)(a) and (b) shows how the accuracy results are converged after five-fold cross-validation. From the cross-validation, it can be inferred that RF and ANN have performed best both in training and testing accuracy. DT performed with approximate similarity to RF and ANN in the case of training. In the case of testing, the DT had most deviations than other techniques and performed poorly at first. However in the last three folds, it performed similarly to RF and ANN. SVM and LR performed weakly than other techniques in training. In the case of testing and in the first two fold, SVM and LR both performed better than other techniques and logistic regression was best among them, but at the last three folds, they performed worse than others. [Table 3](https://www.sciencedirect.com/science/article/pii/S2542660519300241" \l "tbl0003) represents different evaluation metrics for different techniques trained on the dataset. From [Table 3](https://www.sciencedirect.com/science/article/pii/S2542660519300241#tbl0003), it can be seen that DT and RF have more accuracy, precision, recall, and F1 score values than other techniques. ANN also performed well in the case of evaluation. However, DT and RF are a little more accurate than ANN. On the other case, LR and SVM also do well on our dataset but not as good as other classifiers.



(a) Training accuracy for different techniques for 5 fold cross validation

(b) Testing accuracy for different techniques for 5 fold cross validation.

**Evaluation metrics of our study.**

| **Evaluation** | | **Classifiers** | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **Metrics** | | **LR** | **SVM** | **DT** | **RF** | **ANN** |
| Training | Accuracy | 0.983 | 0.982 | 0.994 | 0.994 | 0.994 |
|  | STD(+/-) | 0.0012 | 0.0015 | 0.00081 | 0.00081 | 0.0013 |
|  | Precision | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 |
|  | Recall | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 |
|  | F1 Score | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 |
| Testing | Accuracy | 0.983 | 0.982 | 0.994 | 0.994 | 0.994 |
|  | STD(+/-) | 0.0055 | 0.0064 | 0.016 | 0.014 | 0.021 |
|  | Precision | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 |
|  | Recall | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 |
|  | F1 Score | 0.98 | 0.98 | 0.99 | 0.99 | 0.99 |