

# Leveraging the Machine Learning Tools and Techniques in Enhancing the Effectiveness of Real-time Data Analysis of Internet of Things (IoT) Devices

Updesh Sachdeva

Mount Olympus School, Gurugram, Haryana, India

*<sup>1</sup>Date of Receiving: 28 July 2023, Date of Acceptance: 05 October 2023, Date of Publication: 12 October 2023*

---

## ABSTRACT

The world is evolving quickly, and multiple web-based organizations rely upon gathering information for future analysis. IoT frameworks can access numerous gadgets, and much information can be stored in an IoT cloud like Thing speak. DHT11 gas level sensors will be used in this project to collect real-time data and use machine learning algorithms (random forest, decision tree classifier, linear discriminant analysis) to analyse it. We are looking at the presentation of analyses using measurements like precision, disarray network, accuracy, review, and score to find the best calculation that distinguishes assaults or information peculiarities all the more precisely.

## INTRODUCTION

The IOT represents articles embedded with sensors, registering skills, programming, and innovations that associate and offer information with gadgets and frameworks over the Web without human communication. There has been an increase in the number of cyberattacks targeting Internet of Things (IoT) devices due to their rapid development.

Unapproved clients keep tracking down new ways and provisions for taking advantage of the current IoT Center for unlawful purposes. Machine learning algorithms have been proposed to identify malicious attacks or data anomalies. To identify the presence of attacks, the framework uses three classification-based machine learning (ML) algorithms: Random Forest (RF), Decision tree classifier, and linear discriminant analysis.

## METHODOLOGY

- 1) Configuration of the Hardware: The NODEMCU(ESP8266) Wi-Fi module and the DHT11 Gas sensor are included in the system.
- 2) Software for Arduino IDE: Uploading a basic embedded C program code
- 3) IoT at Thing speak: Sensor-generated data is uploaded to the cloud, and Excel sheets containing data sets are downloaded for further use in algorithms.
- 4) Learning by machine: Jupyter Notebook software is used for the project's machine learning portion. After importing the necessary libraries, the data set is divided into training and testing sets. Different ML algorithms incorporate these sets. The most effective algorithm with the best performance metrics is chosen based on the comparison.

## REQUIRED COMPONENTS

The proposed system requires the following components:

- 1) NODEMCU: NodeMCU is an open-source particularly utilized for IoT-based Applications. This includes hardware based on the ESP-12 module and components that run on Espressif Systems' ESP8266 wifi SoC. ESP8266EX has 17 GPIO pins committed to various capabilities by programming the suitable registers. This part is ideal for IoT-based projects where Arduino is remote. It has an inbuilt Wi-Fi module.

---

<sup>1</sup> *How to cite the article:* Sachdeva U.; Oct 2023, Leveraging the Machine Learning Tools and Techniques in Enhancing the Effectiveness of Real-time Data Analysis of Internet of Things (IoT) Devices; *International Journal of Analysis of Basic and Applied Science*, Vol 7, Issue 4, 6-11

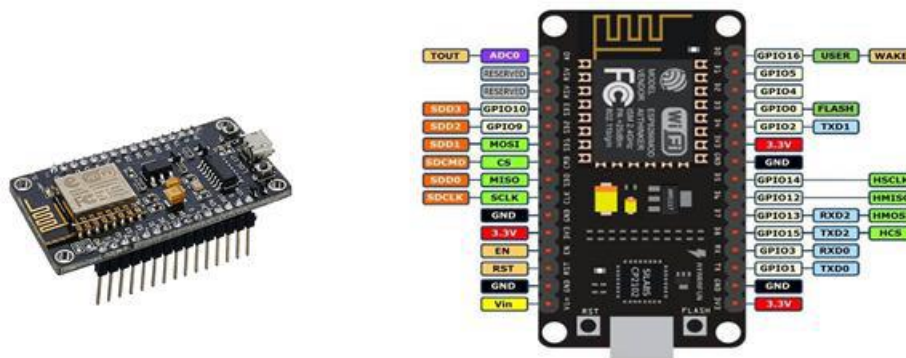


Fig 1: Nodemcu

2) Sensor DHT-11: DHT11 is a low-cost digital sensor for detecting temperature and humidity. It has a thermistor and a sensing element for temperature measurement. A change in capacitance results in the recording of humidity values. Connecting this sensor to any microcontroller, including Arduino, Raspberry Pi, and others, is simple.

To measure moisture and temperature at specific periods. There are four pins on the DHT11 sensor: VCC, GND, Data Pin, and an unlinked pin. Additionally, it has a 5-10k ohm pull-up resistor for microcontroller communication.

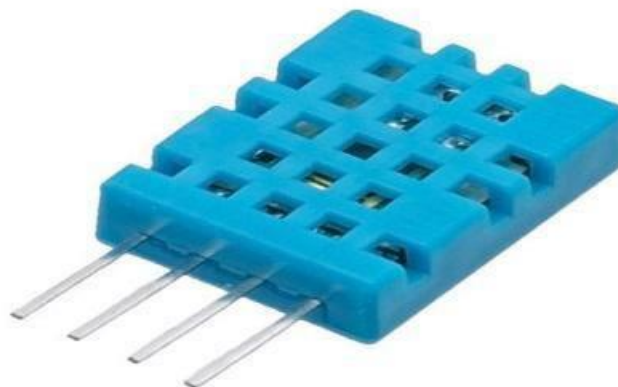


Fig 2: DHT Sensor

3) Gas Sensor: A gas sensor is a type of sensor that measures the concentration or presence of gases in the air. Gas sensors look for potentially hazardous leaks like CNG or petroleum gas to prevent major accidents. There is a sensing element that is a part of every gas sensor. The Electrode, Heater Coil, and Gas Sensing Layer are the components. The sensor's sensing material can identify the gas type. By altering the material's resistance within the sensor, the sensor provides the appropriate change in potential energy response to the gas's concentration level. That adjustment of potential can be named output voltage. The gas's type and concentration level can be assessed based on the voltage value.



Fig 3: Gas Sensor

### WORKING

The gas level sensor and dht11 are used to collect real-time data by the system. These qualities can be shown on LCD, and NODEMCU has an in-fabricated wifi module. The data set can be downloaded as an Excel sheet after it is uploaded to Thingspeak. The Arduino IDE is the system's interface, and the laptop is connected to the NODEMCU cable.

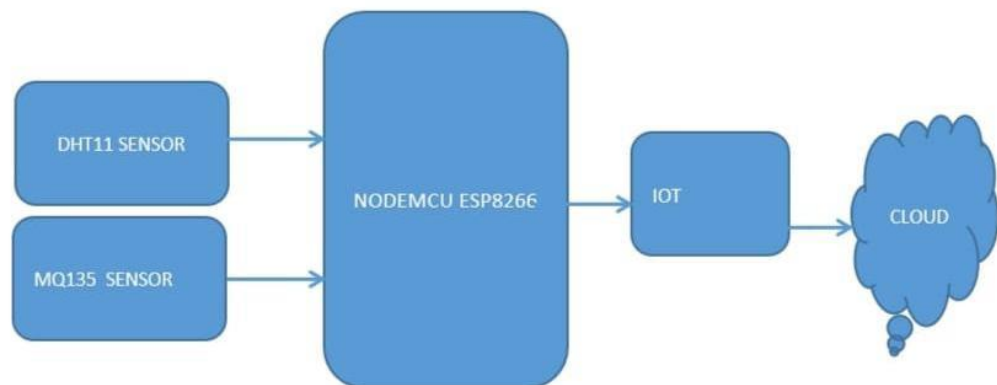


Fig 4: Flow

The machine learning algorithm uses the downloaded data set, which is then converted into a CSV file.

The following are the steps:

- 1) Data sets are posted online.
- 2) The necessary libraries need to be imported.
- 3) Imports of data are made.
- 4) Any missing data must be addressed.
- 5) Encoded categorical data
- 6) Dataset is parted into preparing and test sets.
- 7) Making the model work.

Metrics are used to evaluate each algorithm's efficiency following the model's fitting.

## RESULT

```
Out[33]: DecisionTreeClassifier
DecisionTreeClassifier(criterion='entropy', random_state=0)

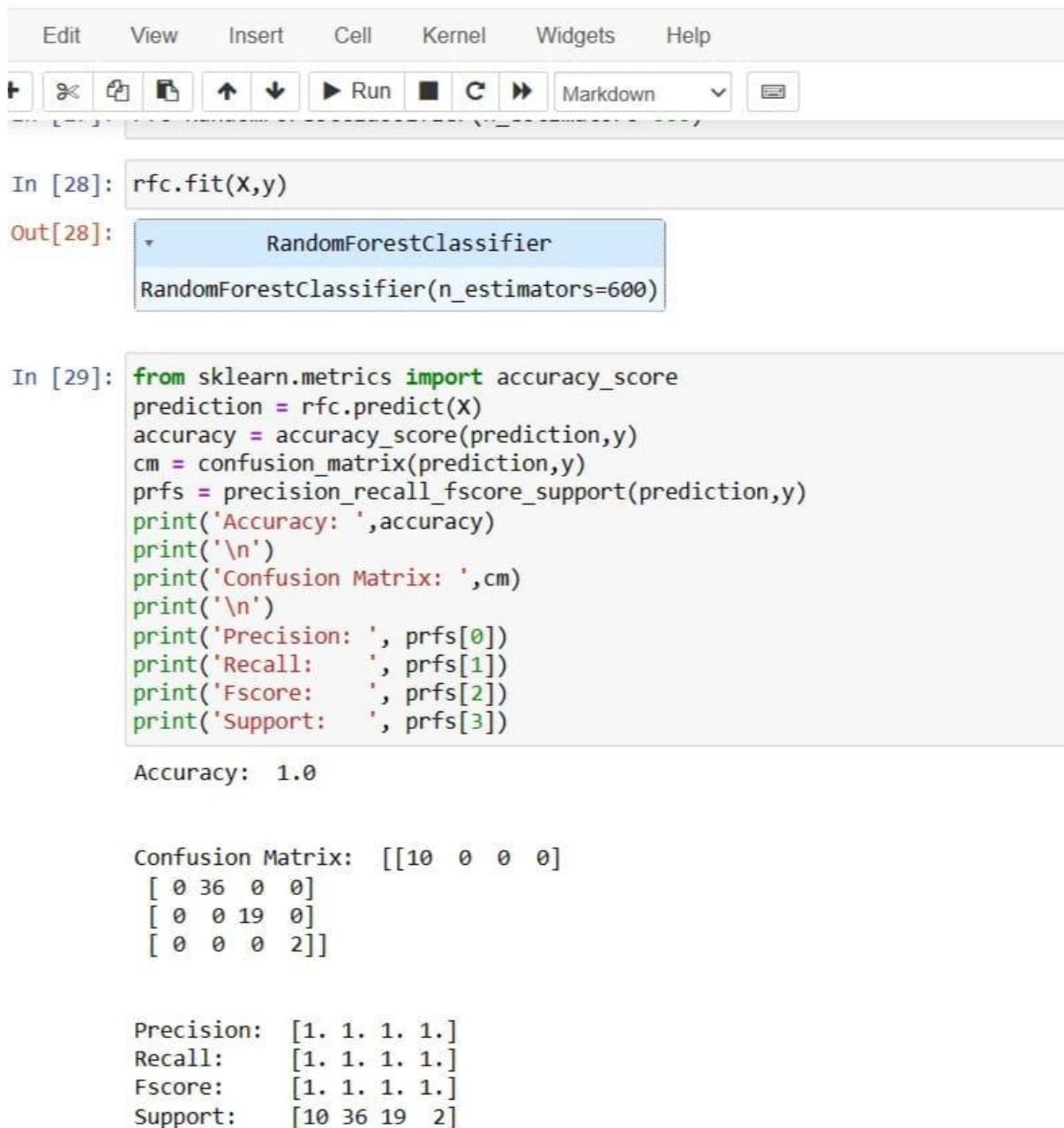
In [34]: prediction1 = classifier1.predict(x_test)
accuracy1 = accuracy_score(prediction1,y_test)
cm1 = confusion_matrix(prediction1,y_test)
prfs1 = precision_recall_fscore_support(prediction1,y_test)
print('Accuracy: ',accuracy)
print('\n')
print('Confusion Matrix: ',cm)
print('\n')
print('Precision: ', prfs[0])
print('Recall: ', prfs[1])
print('Fscore: ', prfs[2])
print('Support: ', prfs[3])

Accuracy: 1.0

Confusion Matrix: [[10  0  0  0]
 [ 0 36  0  0]
 [ 0  0 19  0]
 [ 0  0  0  2]]

Precision: [1. 1. 1. 1.]
Recall:    [1. 1. 1. 1.]
Fscore:    [1. 1. 1. 1.]
Support:   [10 36 19 2]
```

Fig 5. Result of Decision Tree



```

In [28]: rfc.fit(X,y)
Out[28]: RandomForestClassifier
RandomForestClassifier(n_estimators=600)

In [29]: from sklearn.metrics import accuracy_score
prediction = rfc.predict(X)
accuracy = accuracy_score(prediction,y)
cm = confusion_matrix(prediction,y)
prfs = precision_recall_fscore_support(prediction,y)
print('Accuracy: ',accuracy)
print('\n')
print('Confusion Matrix: ',cm)
print('\n')
print('Precision: ', prfs[0])
print('Recall: ', prfs[1])
print('Fscore: ', prfs[2])
print('Support: ', prfs[3])

Accuracy:  1.0

Confusion Matrix:  [[10  0  0  0]
 [ 0 36  0  0]
 [ 0  0 19  0]
 [ 0  0  0  2]]

Precision:  [1.  1.  1.  1.]
Recall:     [1.  1.  1.  1.]
Fscore:     [1.  1.  1.  1.]
Support:    [10 36 19  2]

```

Fig 6: Result of Random Forest

## CONCLUSION

Finding the best algorithm with the best performance for attack detection is the primary objective of our proposed system. After comparing the results, we concluded that the most accurate method is Random Forest. This worth might change depending on the informational collection. It has been determined that random forest outperforms the other algorithms in handling large data sets. The issue arises when Random Forest removes multiple decision trees' combined results.

## REFERENCES

[1] Abeshu, A., Chilamkurti, N., 2018. Deep Learning: The Frontier for Distributed Attack Detection in Fog-to-Things Computing. IEEE Communications Magazine 56, 169–175. <https://doi.org/10.1109/MCOM.2018.1700332>

- [2] Agatonovic-Kustrin, S., Beresford, R., 2020. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research *Journal of Pharmaceutical and Biomedical Analysis* 22, 717– 727. [https://doi.org/10.1016/S0731-7085\(99\)00272-1](https://doi.org/10.1016/S0731-7085(99)00272-1)
- [3] Ali, T., Nauman, M., Jan, S., 2018. Trust in IoT: dynamic remote attestation through efficient behavior capture. *Cluster Computing* 21, 409–421. <https://doi.org/10.1007/s10586-017-0877-5>
- [4] Aminanto, M. E., Choi, R., Tanuwidjaja, H. C., Yoo, P. D., Kim, K., 2018. Deep Abstraction and Weighted Feature Selection for Wi-Fi Impersonation Detection. *IEEE Transactions on Information Forensics and Security* 13, 621–636. <https://doi.org/10.1109/TIFS.2017.2762828>
- [5] Amor, N. B., Benferhat, S., Elouedi, Z., 2021. Naive Bayes vs decision trees in intrusion detection systems. *ACM Press*, p. 420. <https://doi.org/10.1145/967900> 967989
- [6] Bhunia, S.S., Gurusamy, M., 2019. Dynamic attack detection and mitigation in IoT using SDN, in 2017 27th International Telecommunication Networks and Applications Conference (ITNAC). Presented at the 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), IEEE, Melbourne, VIC, Australia, pp. 1–6. <https://doi.org/10.1109/ATNAC.2017.8215418>.
- [7] Wu, M., Song, Z., Moon, Y.B., 2017. Detecting cyber-physical attacks in the cyber-Manufacturing systems with machine learning methods. *Journal of Intelligent Manufacturing*. <https://doi.org/10.1007/s10845-017-1315-5>. 61
- [8] Wang, L. (Ed.), 2020. Support vector machines: theory and applications, *Studies in fuzziness and soft computing*. Springer, Berlin.