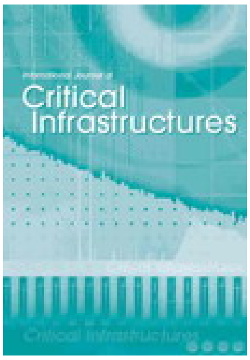
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Critical Infrastructures

ISSN online: 1741-8038 - ISSN print: 1475-3219
https://www.inderscience.com/ijcis

Integrating IoT and machine learning for scalable anomaly detection in smart city infrastructure

Jing Xu

DOI: 10.1504/IJCIS.2027.10074174

Article History:

Received:	28 December 2024
Last revised:	03 February 2025
Accepted:	10 February 2025
Published online:	24 March 2026

Integrating IoT and machine learning for scalable anomaly detection in smart city infrastructure

Jing Xu

Police Tactical Department,
Fujian Police College,
Fuzhou, Fujian, 350007, China
Email: xujing198207@163.com

Abstract: People all over the world can connect a lot of smart things to the internet of things (IoT). These tools can talk to other tools in the same family without any help from people. The internet of things (IoT) lets us get and look at a lot of data. Many good things could come from this. A lot of data is made when more things join the IoT. You might find strange things after reading this. It has a lot of different kinds of things. Standard ways to keep an eye on hacking threats need to handle and process different kinds of data in different ways. This might not work well for files that have a lot of different parts. But data from more than one kind of network gadget can hold more kinds of data. It will help you find strange things more quickly.

Keywords: data pre-processing; anomaly detection dataset; benchmarking models; evolution of anomaly detection techniques; internet of things; IoT.

Reference to this paper should be made as follows: Xu, J. (2026) 'Integrating IoT and machine learning for scalable anomaly detection in smart city infrastructure', *Int. J. Critical Infrastructures*, Vol. 22, No. 10, pp.1–16.

Biographical notes: Jing Xu is an Associate Professor of Police Tactics at Fujian Police College, and graduated from the School of Social History at Fujian Normal University in 2004 and from the School of Sports Humanities and Sociology at Fujian Normal University in 2007. Her research directions are practical police training and multi-dimensional police operations. Since joining the workforce, she has continuously delved into the field of police operations and practical research, publishing over 20 papers, including two in core journals. She has also led or participated in over 20 scientific research projects, including two in national social sciences; two provincial and ministerial level projects, more than ten departmental level projects have been led, and one book has been written, with a solid academic accumulation.

1 Introduction

In the past few years, the internet of things (IoT) has grown a great deal. Groups and people from all over the world now use it. A few of these are smart cities and houses, farms, transportation, business, healthcare, the military and other places. The IoT is connecting more things because of this. The world and people have changed a lot because of this. Adding the IoT to the mix gives you new tools and ways to learn. These have both pros and cons. The IoT has changed many things. Supply 4.0 in operations

(Sundmaeker et al., 2020), smart mobility in transportation (Azmat et al., 2019), and the internet of battlefield things in the military (Ahmed et al., 2021) show how it has changed things. That's the end of it. Also, businesses in this area want to combine IoT technology with other new technologies such as AI (using AI algorithms to process and analyse data), Big Data (handling a lot of data from IoT sensors), or 5G communications (letting IoT sensors move around and talk over fast networks) so they can make smart systems that work together.

A lot of people want to know how AI and IoT can work together. AIoT stands for 'artificial intelligence of things'. One of its main goals is to make IoT networks that can gather information and choose what to do without any help from a person. Still, more and more people from around the world are connecting to the internet. This means that the IoT is quickly getting bigger, more linked, and harder to use. More than ever, people who use the IoT need to stay safe. This also includes the people who use the gear. People who have a stake in the IoT will not use it much until they are sure that their networks and data are safe. A lot of people are scared about their safety because of the IoT. This is because lots of IoT systems are made to be useful, not safe. Hackers are getting into more and more IoT IP settings, according to a new study. They can now link to more IoT devices (De Canditiis and De Feis, 2021). They can now hit the cloud, too. That's why it is so hard for people who work in this field to keep smart things online safe.

Bad people could break into IoT networks and gadgets and add them to botnets. This is how dangerous it is to use phones and the IoT. In the first quarter of 2021, someone writing about DDoS said that IoT users are still a big threat to DDoS e-mails. The whole world has more than 100 of these computers on all the time. They are also known as C2 computers. Some important IoT things make this hard to do. The CPU, memory, and speed are not enough, and it's too simple. It is hard for people who study the IoT' safety to think of new ways to find risks and strange things that happen when it's used (Ahmed et al., 2021). A lot of scholars are interested in this area. You could choose from a lot of IoT files. We chose one that did not have a lot of information on it or known weak spots that could be used to get in. There were tests and tweaks done to make sure it would work with both even and odd groups. This is one way we could find strange things in IoT networks. Last, we looked at how to make this method less likely to fail and how it changed datasets that were even and datasets that were not even.

Anomaly detection (AD) has recently received significant attention. AD refers to the identification of unusual or unexpected patterns in data. To achieve this, it is necessary to analyse a given dataset to determine whether it deviates from expected patterns (Vaidian et al., 2019). There is no universal framework for AD, as its applications vary across different domains. The study of AD is complex, encompassing both general methods applicable to a wide range of scenarios and specialised approaches designed for specific use cases.

This is shown well by the study of intrusion detection systems. Another one looks into AD, how it works, and how IoT systems can be used to guess if someone will get it. Lastly, there was a study that looked at how AD techniques can be used to make flying better and friendlier. This is a great way to teach the subject because it shows the main ideas, problems, and flaws of the different methods and how they can be used in different situations. The whole series changes a lot with this piece of writing. It has points, sets, functions, graphs, and other things in it. At one point, strange things can happen in the world, in people's actions, or a mix of the two. With market data, you can learn and try out different AD tactics. In some ways, this can make AD hard to handle and simple to take care of. AD could give you marks or numbers (Kott et al., 2016). The theme is talked about from more than one source, and two new reviews are also talked about. ‘Novelty detection’ is not the same as ‘anomaly’. ‘Anomaly’ means that the data found are not thought to be normal, but ‘novelty detection’ does not mean that they are not normal.

People say it’s nice when someone shocks them. A new and important study that looks at AD uses big data (Zakariah and Almazyad, 2023). People who work with this kind of data often get the ‘curse of dimensionality’, which stops them from doing many things. There are four main types of AD methods: those that use information theory, those that use grouping, those that use statistics, and those that use classification. Taxonomy is shown in Figure 3, which is a great way to show these four groups. The main subject of this piece is how to use statistics. To begin, use numbers to make a model that fits the information you have. Next, check the data to see if any of them do not match the model. There is a full look at statistical methods, including a range of approaches, such as robust principal component analysis (PCA) methods, robust regression methods, and signal processing methods. In this study, we use statistical signal processing to do our job. Data can be things like people, events, or lengths of time.

It does not matter what you call them: outliers, anomalies, signal processing, functional data analysis (Ahmed et al., 2016), or functional data analysis. All of them mean the same thing. Give this some thought. Some oddities only show up once, while others show up over and over again. People do not mind strange things as much when they are far away. But stubborn people care about a lot of ground. It has not been used much to study Alzheimer’s disease in the past, and when it was, it was mostly for linear functions (Thudumu et al., 2020). Remember this about the books. We’ll look at mixed data, which is another name for different data. This set of data has some points that do not fit with the others. We are going to use skewed multiple regression to look at this subject. They are all kept in a database in the same way, even though they come from K different places. Then, in case something goes wrong, we add a term to each part of the signal. Any of these words could be this one (Rousseeuw and Hubert, 2018; Hubert et al., 2015). When you have the normal form, it’s like having more than one signal. The different parts of the signal will have the same shape some of the time. But some parts might act strangely.

This paper is structured as follows. Machine learning (ML) can be utilised to detect IoT botnets exhibiting anomalous behaviour, which is discussed in Section 2 of the second part of this study. The concept of a ‘robotic brain’ is briefly introduced in Section 3. Autoencoders are of particular importance in this research, as they play a crucial role in later sections. In Section 4, we describe the development, experimentation, and implementation of our autoencoder-based system for detecting anomalies in IoT networks. The results of our findings are presented and analysed in Section 5. In Section 6, we discuss the insights gained from our study, including the unique aspects of our proposed approach. People are working on this idea (De Canditiis and De Feis, 2019). It is shown in a very dense way in another part of this book. This helps figure out what the signal might be and what mistakes are most likely to happen. This kind of mistake can be seen in the rough picture.

2 Literature review

That's because there are now a lot more IoT devices than there were before. They send a lot more when they are tied together. You have to look through a lot of records to find the odd ones. You can find strange patterns or habits this way. The goal of this literature review is to give an outline of the research that has been done so far on how to find problems in active learning systems that use the IoT. A lot of different ideas have also been put forward in this area. Because of this, they cannot have a job that only lets them attack. People's safety systems use ML to find flaws and risks in the way they work. You can use a tool that only works with smart things instead. It could work better and be more useful if you do this. To do their work, they used the NSL-KDD and UNSW-NB15 tools. Its F1 score, how long it takes to learn, how long it takes to test, and how well it remembers things show how well it works. The F1 score for the RFs is 0.99 (Alsaedi et al., 2020), and the F1 score for the SVM is 0.65 (Louk and Tama, 2023). The main goal of this research was to look into the various types of hits that can happen.

It was rare for things used to find hits to go off on their own. A lot of different types of songs did well. A way to find strange things was looked into to make a smart city safer, according to the study. Experts used a decision tree (DT), a K-nearest neighbour (KNN), and radio frequency (RF) to make the smart city's infrastructure safer and lower the risk of possible dangers. The study's writer's look into how to make the design's spotting better by using group methods like boosting and bagging. They want to know how to use two sets of data: CICIDS 2017 and UNSW-NB15 (Krishnaveni et al., 2022). The SVM did its job well because 90% of the time, it got the answers right. The artificial neural network got it right 79.5% of the time, the boosting method got it right 98.6% of the time, and the stacking method got it right 98.8% of the time. The test results from the UNSW-NB15 dataset could help us find threats that do not happen very often in a smart city IoT setting. It also talked about a new way to find threats that fog nodes can be used for. We found web of things traffic that was not needed with the help of the UNSW-NB15 features. This study also has the tab translate model that you can look at.

It's better than AI. This model should be able to tell the difference between normal traffic and strange traffic 98.35% of the time. Their method must work because it hits 97.22% of the time for a lot of different types. The model has given us new ways to look into fog node problems (Zhou et al., 2020) also talked about a number of ways to find security holes. They learned how to use the UNSW-NB15 dataset as they worked on it (Patrikar and Parate, 2022). We tested naive Bayes (NB), LR, KNN, and RF to see how well they could find attacks. We tried different ways to group things, both with and without feature selection. The error rate, the speed, the precision, and the memory were the things that were measured. These were the F1 score, the MSE, the FPR, the TPR, and the true positive rate. It was also possible to compare these machine-learning models with each other. When you add up all the data, the RF method is the most accurate because it works 99.5% of the time. If only certain traits are used, it's 99.6% likely to work (Hassan et al., 2022) The UNSW-NB15 dataset's features were used to find groups of features based on flow, TCP, and message queuing telemetry transport (MQTT).

It did not matter if they were too big or too small or if the collection was not balanced. The groups were taught using ANN, SVM, or RF, which are guided machine-learning methods. They were right: 98.67% of the time when they used RF-based binary classification and 97.37% of the time when they used multi-class classification (Chowdhury et al., 2007a). When they used the best features from each group – RF on flow features, MQTT features, and TCP traits – it worked 96.74% of the time, 91.96% of the time, and 96.56% of the time. The most advanced directed ML methods work better and take longer to train than some feature groups.

2.1 Evolution of anomaly detection techniques

There were many ways for people who liked to find strange things to work together and do their jobs. When people made things by hand, they often found weird things. Along with the trajectory-based method and the histogram of oriented gradient (HOG), these are some of the other methods in this group. Neuron-based computers can now find their way and tell time on their own thanks to progress in ML. Because neural networks got better, the gradient that went away had to be fixed. A deep sparse rectifier neural network was thought of as a way to fix the fading gradient glitch in 2011. This network would work because of ReLu. This was one of the most important steps forward in the history of neural networks. CNNs were made so people could learn about time and space on their own because it was so important. Since 2012 (Chowdhury et al., 2020b), CNNs have made it easy to fix a lot of issues with the way pictures are made. They look for things that do not make sense. It is made up of stacked perceptrons, which are a type of brain cell.

Also, it has one or more convolutional layers that can be fully linked together or shared. You can save a picture piece on a feature map. These are made with convolutional layers. These feature maps are not sent to be worked on right away after they have been fixed up. Deep convolutional neural networks are one way to learn more and get better ideas. There are many levels in these networks that you cannot see right now. A lot of the time, envelope – decoder models are used for deep learning. People who encode data learn how to draw it, and people who decode data use this picture to try to figure out what the data is (Chriki et al., 2021). When you encode and decode, you can see how simply the raw data is spread out. It helps you figure out what’s wrong. Some of them do these things differently, though. CNN does not always tell the truth. A recurrent neural network (RNN) can easily handle input that comes in a set order over time.

CNNs are designed to work with pictures, while RNNs are designed to work with text and voice. CNNs work faster than RNNs because they are better at handling pictures. RNNs cannot tell the difference between things that look like they are the same but are not.

LSM neurons remember things for a short time (RNN) and are used to find strange things. An LSTM network is a type of RNN. LSTM networks are different since they have a memory cell that can keep data for a long time (Chriki et al., 2020). Long short-term memory (LSTM) can figure out what patterns in data mean by looking at them. It is simple to discover strange things now (Ullah and Mahmoud, 2020). The last few years have seen deep neural networks (DNNs) get very good at handling a lot of data, mostly pictures. Some tools can be used to find pictures, group them, and make new ones. It’s still hard to make pictures that look real and have a lot of details. Its number (Chu et al., 2018), they did not get any instructions, so they made a model that could learn any data spread. Some people have looked at the one-class classification (OCC) method. This way of searching looks for things that are in a certain group. To find the hypersphere with the fewest points, OCC is often used. An SVM is a useful machine that this method is based on. The OCC wants the first line to be as long as it can be because they do not know anything about the negative class. OCC (Lai et al., 2024) can be split into three more groups as well. Both types use generative adversarial networks to make edges. The loss function is used to tell the difference between two things in the second type. OCC is mostly used to find strange and new things.

3 Methodology

There should only be one way to test ML models. You should only use this method in a place with an online tracking network. This is how you can find out about strange hacking events. That way has a tool that looks for problems. We also look closely at how well different ML models work in general and how they were made. We use the best hyperparameters we can find to see how the models work. This way, we can see how well different simple models work. This helps us find ML models for IoT that work well everywhere (Apostol et al., 2021). To find the best hyperparameters, we use the Bayesian method. Bayesian optimisation (BO) tries to find the function's best value. It aids in making plans that follow a straight path. The machine needs to find ones that are different as quickly as possible. Planning what to do is what Bayesian planning is all about. Next, we present a group of ensemble models that can take the best parts of several less powerful prediction models and put them together to make a single model that can make better predictions. After that, it talks about how the info was cleaned up and set up.

It also talks about feature engineering and how the study's different models were put together. Here's what we talked about today in terms of how things really work. This kind of thing is called 'experimental results'. We will talk about what we think are the most important factors and how well the model works in 'hyperparameters optimisation'.

3.1 Data pre-processing

First, we check the whole record to make sure it's the same. First, you need to find the set's mean. Next, you need to give it a number of units. Drawing the data point will help us understand it better. We make sure that each trait becomes a

$$X'_{i,j} = \frac{X_{i,j} - \bar{X}_j}{\hat{\sigma}_j} \tag{1}$$

where

$$\bar{X}_j = \frac{1}{|x|} \sum_{i=1}^{|x|} X_{i,j}, \tag{2}$$

$$\hat{\sigma}_j = \sqrt{\frac{1}{|x|} \sum_{i=1}^{|x|}} \left(X_{i,j} - \bar{X}_j \right) \tag{3}$$

Make the number of cards in the part stand out. We also find the Pearson relationship value for each set of traits we were given. Here's how to get the link between the number and the:

$$r_{j,k} = \frac{\sum_{i=1}^{|x|}\left(x_{i,j} - \overline{x}_j\right)\left(x_{i,k} - \overline{x}_k\right)}{\sqrt{\sum_{i=1}^{|x|}\left(x_{i,j} - \overline{x}_j\right)\sum_{i=1}^{|x|}\left(x_{i,k} - \overline{x}_k\right)^2}} \tag{4}$$

After that, we can check how strong the link is between the two traits. If this comes out to be true, we will take away the tool to keep things from getting out of hand. A wall makes it impossible to have traits that are strongly linked. We looked at this amount for our job. All category traits except IP address traits are combined into a single hot-encoded feature. All the time there are almost 5,000 different groups. It changes a lot, but it does not always help us figure out what will happen.

3.2 Anomaly detection dataset

For each basic model, some private data is used to test it. This is what we need to do to make a model that works well in a number of IP datastream fields. Let's quickly go over what each file contains and the types of things that are in them. A full network record was not what the IoTID20 was meant to be. It is actually a set of attacks for smart things. It is partly based on flow. With these new flow-based botnet traits, we can now look at systems that look for risks. This is a great way to keep an eye on networks that are acting strangely for the IoT. NUGU (NU 100) and EZVIZ Wi-Fi Camera (C2C Mini O Plus 1080P) send attack packets to it. Attack texts came from phones and computers that were all connected to the same Wi-Fi network. All of them are in the file. You can find out more about eighty different network groups and features at IoTID20. Besides the named classes, the group has three other types:

Two kinds of units can use the 1 IoTID20 binary. Links that work do not do anything bad or strange. II. Bad for the web: this means that hacked tools are giving out information that is not good for them.

Many groups make up the 2 IoTID20 Multi-Cat. It can be split into five groups. The normal dataset and the binary dataset both show the same flow data. Then, they will not be able to change things that are shared. A 'DoS' attack. A 'Man in the Middle' hack can use code from a network. This is called MITM ARP Spoofing. Someone who knows about the network can trick an ARP server. Something turns into a bot that you can control from afar when you put Mirai software on it. The stream that looks for IoT devices is named 'scan'. There are nine groups of risk information that the IoTID20 Multi-SubCat tool sorts into. DOS-Synflooding is an attack method that uses the SYN flood. The ones that came before these are the same as these. The MITM ARP fake attack is a type of Man in the Middle attack that is used with the ARP scam. It's called mirai-ack flooding when the Mirai Bot sends a lot of TCP ACK packets. A lot of HTTP requests go out at the same time. It is called an HTTP flood. The Mirai Bot tells a fake machine something in this case.

This is called 'brute forcing'. These are sent by something or someone called the Mirai Bot. This is known as a Mirai-UDP flood. This is the first type of data. It checks the address for open ports. The second type, scan port OS, looks for open ports on the work system. Getting network data from IoT-23 is a new idea. IoT gadgets in smart homes send a lot of data to it, which it saves. The Amazon Echo, the Phillips HUE, and the Somfy Door Lock can all record. Every day, people use software and the IoT. The line shows real threats to these things. So far, IoT-23 has heard of 23 of them. Twenty of them are bad network traffic, and the last three are good traffic logs. Besides that, it used old technology that computers today have never seen before. This tells us about the newest technology and how safe it is with the safety features we already have. Each class's boxes with less than five items are thrown away. Then, we load the first million records for each case and put people whose names do not match into groups based on their names. Now, we have a clean sample.

There are two groups of named classes in the set: the 4 IoT-23 binary can be joined to two of these. This kind of network info might be good or bad. A lot of people click on links, but they never do anything bad or strange. This type of journey is called 'innocent traffic'. Tools know they should not do bad things, but they still do them. The IoT-23 Multi-Cat can be used with two different types of classes. This kind of network info might be good or bad. A lot of people click on links, but they never do anything bad or strange. This type of journey is called 'innocent traffic'. Tools know they should not do bad things, but they still do them.

Table 1 summarises all examined data. Visit the site to get our empirical study dataset and labels. Below, we explain this study's models and Bayesian hyperparameter optimisation.

Table 1 A detailed overview of the distribution of classes within the evaluated dataset

Dataset		*Label*		*Train set*	*Test set*	*Label instances*
IoTID20	Binary	Normal		31,979	8,094	40,073
		Anomaly		468,353	116,989	585,342
			Total	500,332	125,083	625,415
IoTID20	Multi-Cat	Normal		31,979	8,094	40,073
			Total	500,332	125,083	625,415
IoTID20	Multi-Subcat	Normal		31,979	8,094	40,073
		DoS-Synflooding		47,537	11,854	59,391
			Total	500,332	125,083	625,415
IoT-23	Binary	Benign		1,462,947	366,180	1,829,127
		Malicious		9,099,261	2,274,373	11,373,634
			Total	10,562,208	2,640,553	13,202,761
IoT-23	Multi-Cat	Benign		1,462,947	366,180	1,829,127
		PartOfPortScan		5,966,736	1,492,481	7,459,217

3.3 BO with tree-structured Parzen estimator (TPE)

In this study, the TPE is used to teach our fake model. The TPE method builds models one after the other to see how well hyperparameters work based on data from the past. This method uses BO with a sequence model. It will then change its model and pick a new set of hyperparameters that it thinks will help it run faster. This is the right way to describe hyperparameter optimisation. You can use this map to see how well the model's hyperparameters x work. All of the possible hyperparameters make up the real world. The goal of BO optimisation is to get the target score as low as possible, as shown in figure *f*(*x*). We are going to talk about the hyperparameters that can give the worst results, as shown by

$$x^* = \arg\min_{x \in X} f(x) \tag{5}$$

What is the name of the test that this study uses to see if it worked? This is where we look for IoT risks; *y* is the usual F1 number. We can then change the goal of the BO optimisation to get rid of as many bad F1 scores as we can. This is how we measure success. What is the chance that the goal will be scored? We need to check the game's rules. We do not want to change the parameters directly, so we change how hyperparameters are made and how they are spread out with densities that are not based on the parameters. This is done in TPE. The first step in the TPE process is this. A bunch of different hyperparameters were used to make some notes. To sort the data, TPE first finds the objective score for each item. Then, it puts the data into groups based on those scores. What do quantiles do for you? You can plan your data and find out how dense the numbers are with Parzen estimators. The old hyperparameters are then used to make the new ones. Most likely to be changed are the ones that are sent back. It is possible to use hyperparameters that have already been looked at to make a statistical model with TPE. After that, it tells you what hyperparameters to check next.

3.4 Benchmarking models

This part has a lot of the tests that were done on the machine-learning models for each dataset. Today, we looked at 14 different kinds of ML models. Ten people played the parts. There were 16 single models and eight group versions. The hyperparameters that make a model work are different for each one. With a hierarchical Gaussian process and a tree-structured Parzen guess, it will take 45 tries to get the best set of numbers. Click on the 'hyperparameters optimisation' part to see a picture of how the hyperparameters change the model.

Our benchmarking models are as follows.

1 The factors can only be a certain number, but this method still sorts things into groups. This is called the ridge regressor (ridge), which is also known as ridge. Ridge views the setting as a regression process; when there are more than one class, there are many different outcomes. It can learn quickly from many things because its line is straight. It can only guess what will happen when all the facts are known.

2 Naive Bayes: you can use Bayes theory to find the prior chance for each class and see how dense each class is. Based on the test, Bayes theory is used to figure out what the chances are for each group. Then, it is given to the group whose posterior class probability is the highest. NB is also great because it can quickly learn from lots of records. In NB, there is not much choice, but in real life, that does not work very often.

3 A feed-forward neural network is known as MLP1. Backpropagation helps it learn new models. Because it changes the weights between neurons, MLP can guess better most of the time. MLP can read text and find patterns and trends that are hard to understand most of the time. MLP gets stuck there a lot of the time, and it takes a

long time to fix. On the other hand, it can do well with news that it has not seen before.

4 Support vector machine (SVM): Drawing a line through each piece of data is an easy way to divide it into groups; because of how its engine works, the help vector machine (SVM) can sort jobs that are not smooth. The input features are turned right away into feature areas with more measurements when this method is used.

5 It looks like a forest, which is another way to write 'DT'. It is a way to make decisions without following rules. It has a lot of nodes and stems. There is a decision rule for each trait on each line. How come this works so well? DT is easy to understand since it is based on rules.

6 K-nearest neighbour algorithms only keep the traits you give them after the sixth level. The facts are then put into groups using a measure of how close they are to each other. kNN is used in many places because it is a simple and non-parametric way to find things.

7 With this method, a random piece of data is used to teach a lot of weak models. This method is also known as 'bagging'. For the bagging model, you can also find the answer by adding up the guesses that its weak internal categories have made. If you have too many different things in your collection, adding bags can help calm it down. If you use bagging techniques instead of boosting methods, you can learn each one separately. Datasets with a lot of different data points might not fit too well when bags are used.

8 That's another name for the eight adaptive boosting methods. It is a meta-classifier that can learn new things from many places. AdaBoost's best guess comes from adding up what people who were not thinking about it themselves said would happen. A weighted sum is used for this. It would be even better if they put more weak kids in with tough ones. Because of this, older systems do not mess up as often.

9 It takes the best scores from several DTs and combines them to make one guess. This is known as the RF method. RF gives each DT a different set of data to learn from. To make the bootstrapping method work, rough traits are used. Then, to change them, data features are added to these features.

10 A group method that does the same thing is both the random forest (RF) method and the ten extremely randomised trees (ERT) method. It teaches a lot of weak DTs how to learn badly. Choice trees look at all of the data, while RF only looks at a small part of it. ERT also picks the split point at random to separate the nodes in the DT. RF, on the other hand, knows ahead of time which split will work best. ERT can learn faster than RF because the nodes are spread out. At first, it does not change much else.

11 This is one way to write the gradient boosting machine (GBM): a lot of people work together to make a lot of bad plans, which are then put together one by one. A lot of people in Britain use a choice tree when the weather is bad. That is the same as RF. Because each weak predictor is taught to fix the residuals of the one before it, RF has a bigger model bias than GB. This is why GB works better than RF.

12 It is based on the gradient boosting method. Extreme gradient boosting (XGB) is a way to give trees more strength. To find the best constant at each place in GB, first-order derivatives are used. In XGB, second-order versions are used instead. It also makes the tree look more normal, which is not quite right.

13 People vote to find out more in 13 voting, and the results that different models say will happen are not the same. It's their choice. They can pick the most options or try to name the classes correctly most of the time. People who guess at random can join the group that votes. These also help fix problems with stacked models most of the time.

14 This method, which is also called 'stacked generalisation', lets you guess what will happen in more than one way. It puts together the results of all the models and then guesses based on the best one. It's simple to stack, and you can quickly mix various model types. In other words, the good things about some models can be used to make other models better.

4 Analysis

Data mining and ML are used a lot to make systems that watch out for strange people and things. A big part of data mining is learning how to look at data, put it in groups, and show it. A lot of the time, the info is sorted for you immediately. It's important to put things into groups that are as good as they can be. A model is made up of groups of things' parts. To find new threats, things that are already used for ML need to be made better. Yes, this is correct since attack styles change all the time. Between 83 and 3 label traits make up the IoTID20 set. There are several ways to figure out how good a guess is. A lot of the time, people use tools like the F-measure, accuracy, precision, and memory.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F\text{-}measure = \frac{2\ Precision \cdot Recall}{Precision + Recall} \tag{9}$$

Various cross-validation tests can be used to check how well a model works in the field of ML. One kind of test that can be used is K-fold cross-validation. There are also separate tests and jackknife tests. This test takes a long time to run because some of the data sets are very big. It still does what it says it will do. We used several K-fold cross-validation tests to speed up the process and find out how well the different groups did. There were checks done on the IoTID20 tags, groups, and sub-tags to see if they had names. These lines show how the traffic is moving. These lines show the road's strange flow. The SVM, GaussianNB, LDA, logic regression, DT, RF, and ensemble models were used to make computer programs that can learn on their own.

Figure 1 Learning curve for labels, categories, and subcategories (see online version for colours)

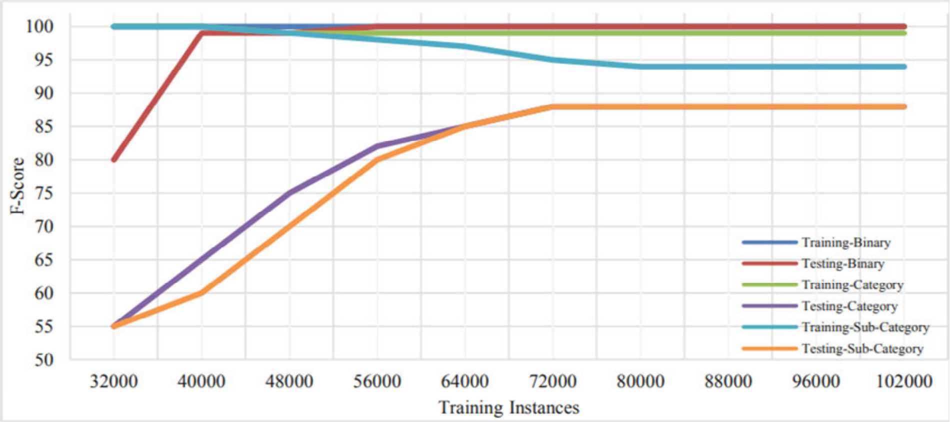
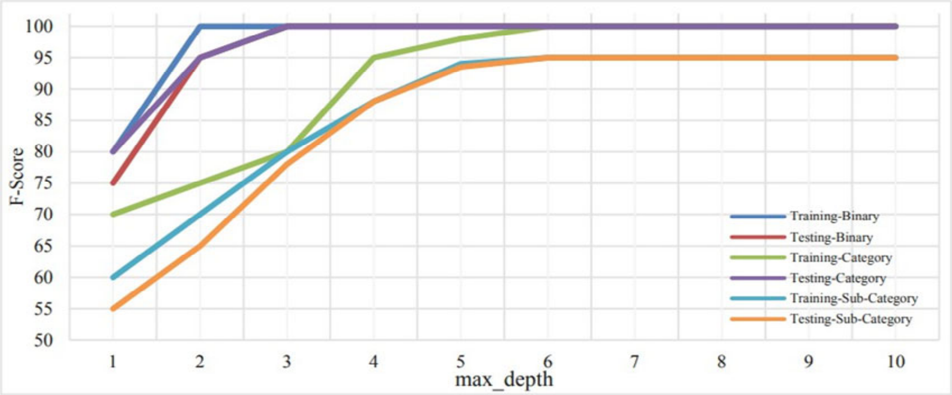


Figure 2 V label, category, and subcategory validation curve (see online version for colours)



That's possible. There are different ways to teach and test a program. The learning curve shows how the two are connected. You can use this graph to see if the program has enough data to work better or if you need to add more data. You can see the F-score learning curve for label sorting into binary, category, and subcategory in Figure 2 for the DT method. Because it shows how well you remember things, the F-score was used to find the slope. The DT needs to see at least 70,000 cases before it can really get good at putting names into lists, categories, and subcategory groups. The IoTID20 dataset was learned with Gaussian NB, LDA, logic regression, RF, SVM, and ensemble models in interesting ways. To get a better score, at least 7,000 pieces of info were needed.

What does a line of trends show? It tests how well it works with both new data and the data it learned from. These are the things you need to get the best score on your work. When the validity curve was made, the max_depth of the DT was set to 10. Figure 2 shows that for label, category, and subcategory names, that is either true or wrong. The classifier score got high enough when max_depth hit two so that everyone could agree on a 2D name. That's when max_depth hit six. The program did so well that it was able to fix the names of both the main and parts. This guess does a good job of putting binary and subcategory names in the right group. Things are not put in the right group very well.

Figure 3 Binary label F-score (see online version for colours)

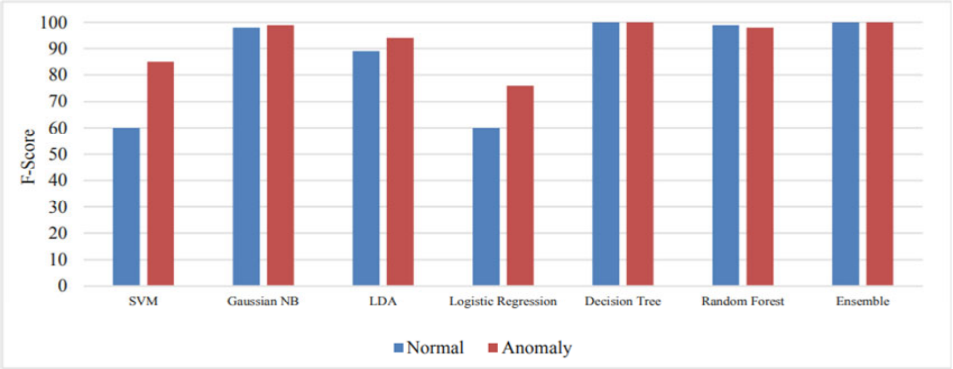


Figure 4 Binary label F-score (see online version for colours)

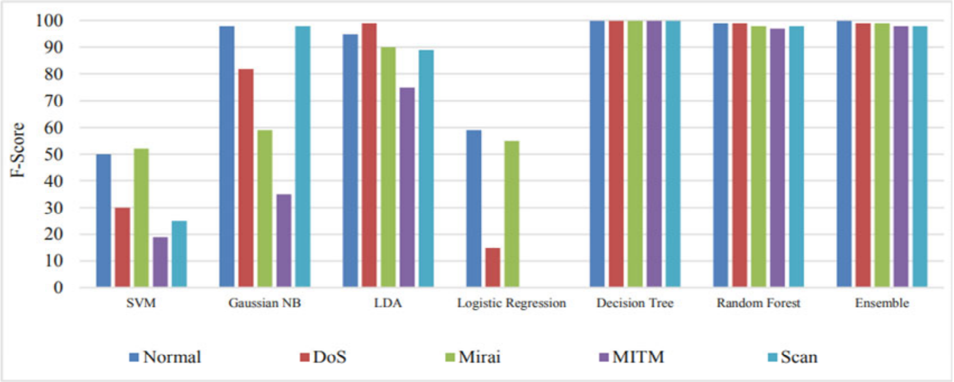
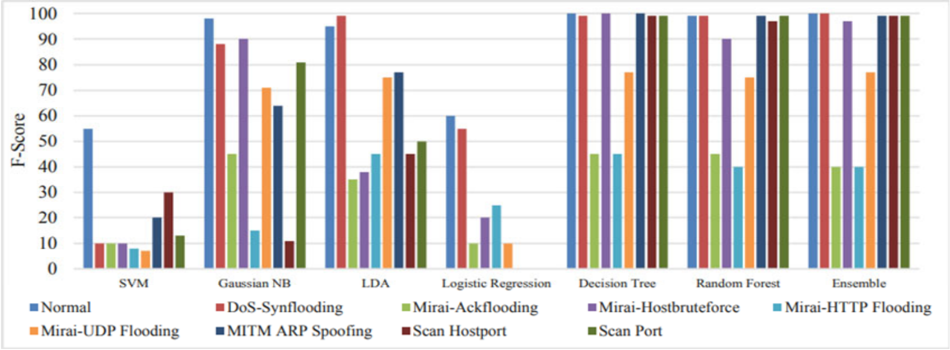


Figure 5 F-score for subcategory label (see online version for colours)



4.1 Binary classification

The flag lets us know if the dataset has examples of networks that do not work or ones that do. No matter how they were used, logic regression, SVM, Gaussian NB, and LDA could not separate things into two groups with different names. However, the ensemble, DT, and RF algorithms all did a great job. Number 7 shows the F-scores for the different ways this study was carried out. We checked these models against the IoTID20 dataset three, five, and ten times to see if they fit too well. After the cross-fold test, it did not change.

4.2 Category classification

No matter what the title says, we could talk about DoS, Mirai, MITM, or scan attacks. You can protect yourself from any attack with the choice tree. That being said, logic regression, LDA, and SVM did not do well with this group of threats. The cross-fold was checked with tens, threes, and fives. The models were then checked to see if they were too big. The cross-fold confirmation study found the same things. How well did you string the subject names together? You can find your F-score by looking at Figure 5. 4.3% is that. Putting the pieces together in groups, Figure 5 shows nine groups of data. The information can go in any of them, or it can just be called 'network traffic'. The best way to deal with the subgenres was with our choice tree method. However, the logic regression, LDA, Gaussian NB, and SVM models were hard to hit with these kinds of attacks. Figure 5 displays the F-score for the subcategory name in the IoTID20 dataset. To help them get better at accuracy, precision, memory, and F score, we wanted to make a ML model. This is because some risks were put in the wrong group. These things belonged to the IoTID20 group. The IoTID20 group's average truth value, accuracy, memory, and F-score can be seen in Table 2.

Table 2 Performance of IoTID20 dataset

Algorithm	*Accuracy*	*Precision*	*Recall*	*F Score*
SVM	40	55	37	16
Gaussian NB	73	70	66	62
LDA	70	71	71	70
Logistic regression	40	25	39	30
Decision tree	88	88	88	88
Random forest	84	85	84	84
Ensemble	87	87	87	87

5 Conclusions

It is useful to compare different ML methods with IoTID20 datasets. To help people find networks that do strange things, a brand-new list was made: IoTID20. There are many types and groups of IoT threats on this list. We looked at some more attack detection datasets and talked about some bad things that could happen. We've thought of eight types of risks. You can test these to see how well IoT network systems that watch out for hackers work. You were also given a list of traits that go well with each other. This list can help you sort the traits and choose the most important ones. Seven types of ML were used to look at the data. The best number that can be achieved with ensemble algorithms and DT algorithms is tied to these three scores: Pr, Rc, and F1. Now that you know what we taught you, there will be a new way for you to find bad things in IIoT networks. We want to build and test a system for models that can find networks that do not make sense as part of our work going forward.

Acknowledgements

Fujian Provincial Department of Science and Technology University Industry University Research Project 'Smart community three level grid multi-dimensional visualization comprehensive prevention and management system' (No. 2023Y4020).

References

Ahmed, M., Mahmood, A.N. and Hu, J. (2016) 'A survey of network anomaly detection techniques', *J. Netw. Comput. Appl.*, Vol. 60, No. 1, pp.19–31.

Ahmed, S., Kalsoom, T., Ramzan, N., Pervez, Z., Azmat, M., Zeb, B. and Rehman, M.U. (2021) 'Towards supply chain visibility using internet of things: a dyadic analysis review', *Sensors*, Vol. 21, No. 6, p.4158.

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A. and Anwar, A. (2020) 'TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems', *IEEE Access*, Vol. 8, No. 9, pp.165130–165150.

Apostol, I., Preda, M., Nila, C. and Bica, I. (2021) 'IoT botnet anomaly detection using unsupervised deep learning', *Electronics*, Vol. 10, No. 16, p.1876.

Azmat, M., Kummer, S., Moura, L.T., Gennaro, F.D. and Moser, R. (2019) 'Future outlook of highway operations with implementation of innovative technologies like AV, CV, IoT and Big Data', *Logistics*, Vol. 3, No. 6, p.15.

Chowdhury, S.S., Islam, K.M. and Noor, R. (2020a) *Anomaly Detection in Unsupervised Surveillance Setting using an Ensemble of Multimodal Data with Adversarial Defence*, arXiv preprint arXiv:2007.10812.

Chowdhury, S.S., Islam, K.M. and Noor, R. (2020b) *Unsupervised Abnormality Detection using Heterogeneous Autonomous Systems*, arXiv preprint arXiv:2006.03733.

Chriki, A., Touati, H., Snoussi, H. and Kamoun, F. (2020) 'UAV-based surveillance system: an anomaly detection approach', in *2020 IEEE Symposium on Computers and Communications (ISCC)*.

Chriki, A., Touati, H., Snoussi, H. and Kamoun, F. (2021) 'Deep learning and handcrafted features for one-class anomaly detection in UAV video', *Multimed. Tools Appl.*, Vol. 80, No. 2, pp.2599–2620.

Chu, W., Xue, H., Yao, C. and Cai, D. (2018) 'Sparse coding guided spatiotemporal feature learning for abnormal event detection in large videos', *IEEE Trans Multimed.*, Vol. 21, No. 1, pp.246–255.

De Canditiis, D. and De Feis, I. (2019) 'Simultaneous nonparametric regression in RADWT dictionaries', *Comput. Stat. Data Anal.*, Vol. 134, No. 6, pp.36–57.

De Canditiis, D. and De Feis, I. (2021) 'Anomaly detection in multichannel data using sparse representation in RADWT frames', *Mathematics*, Vol. 9, No. 11, p.1288.

Hassan, I.H., Abdullahi, M., Aliyu, M.M., Yusuf, S.A. and Abdulrahim, A. (2022) 'An improved binary manta ray foraging optimization algorithm based on feature selection and random forest classifier for network intrusion detection', *Intell. Syst. Appl.*, Vol. 16, No. 11, p.200114.

Hubert, M., Rousseeuw, P. and Segaert, P. (2015) 'Multivariate functional outlier detection', *Stat. Methods Appl.*, Vol. 24, No. 7, pp.177–202.

Kott, A., Swami, A. and West, B.J. (2016) 'The internet of battle things', *Computer*, Vol. 49, No. 11, pp.70–75.

Krishnaveni, S., Sivamohan, S., Sridhar, S. and Prabhakaran, S. (2022) 'Network intrusion detection based on ensemble classification and feature selection method for cloud computing', *Concur. Comput. Pract. Exp.*, Vol. 34, No. 5, p.e6838.

Lai, T., Farid, F., Bello, A. and Sabrina, F. (2024) 'Ensemble learning-based anomaly detection for IoT cybersecurity via Bayesian hyperparameters sensitivity analysis', *Cybersecurity*, Vol. 7, No. 1, p.44.

Louk, M.H.L. and Tama, B.A. (2023) 'Dual-IDS: a bagging-based gradient boosting decision tree model for network anomaly intrusion detection system', *Expert Syst. Appl.*, Vol. 213, No. 3, p.119030.

Patrikar, D.R. and Parate, M.R. (2022) 'Anomaly detection using edge computing in a video surveillance system', *International Journal of Multimedia Information Retrieval*, Vol. 11, No. 2, pp.85–110.

Rousseeuw, P. and Hubert, M. (2018) 'Anomaly detection by robust statistics', *WIREs Data Mining Knowl. Discov.*, Vol. 8, No. 3, p.e1236.

Sundmaeker, H., Guillemin, P., Friess, P. and Woelfflé, S. (2010) *Vision and Challenges for Realizing the Internet of Things*, Clust. Eur. Res. Proj. Internet Things Eur. Commission, Vol. 3, pp.34–36.

Thudumu, S., Branch, P., Jin, J. and Singh, J.J. (2020) 'A comprehensive survey of anomaly detection techniques for high dimensional big data', *J. Big Data*, Vol. 7, No. 12, p.42.

Ullah, I. and Mahmoud, Q.H. (2020) 'A scheme for generating a dataset for anomalous activity detection in IoT networks', in *Canadian Conference on Artificial Intelligence*, Springer International Publishing, Cham, May, pp.508–520.

Vaidian, I., Azmat, M. and Kummer, S. (2019) *Impact of Internet of Things on Urban Mobility* [online] https://www.innovationarabia.ae/wp-content/uploads/2020/10/IA-12-Proceedings-Health-and-Environment.pdf#page=4 (accessed 7 June 2021).

Zakariah, M. and Almazyad, A.S. (2023) Anomaly detection for IoT systems using active learning', *Applied Sciences*, Vol. 13, No. 21, p.12029.

Zhou, X., Hu, Y., Liang, W., Ma, J. and Jin, Q. (2020) 'Variational LSTM enhanced anomaly detection for industrial big data', *IEEE Trans. Ind. Inform.*, Vol. 17, No. 9, pp.3469–3477.