UNIVERSITY OF ALBERTA

Machine Learning and its application in IoT

MINT 709 CAPSTONE PROJECT REPORT

Submitted By:
Sai Vamsi Kolla

MASTER OF SCIENCE IN INTERNETWORKING

Department of Computing Science

Under the guidance of
Juned Noonari

DECLARATION

Sai Vamsi Kolla declares that this original work was completed on ML and its application in IoT in the Department of Computing Science, Master's in Internetworking, University of Alberta. Also, it should be indicated that this study will not be sent to any other educational establishment.

SAI VAMSI KOLLA

ACKNOWLEDGEMENT

The satisfaction in succeeding in this endeavour is not complete without mentioning the people who made it possible and whose never-ending guidance uplifted every effort successfully.

In the progress of this project, a massive investment of time was made. Various influential people gave the project their time and resources. I thank them for their guidance.

I thank **Mr. Juned Noonari** for continuously guiding me in completing this capstone.

My sincere appreciation to our Program Director, Dr. Mike MacGregor, for giving me this fantastic opportunity to experiment with this project.

I want to thank my relatives, colleagues, and friends for their valuable patience, caring thoughts, and encouragement.

SAI VAMSI KOLLA

ABSTRACT

Internet of Things has a tremendous economic and social impact on our lives. IoT links many smart objects. Massive amounts of IoT systems-generated data are typically resource-constrained. More IoT applications and services are developing every day. Any significant contribution to the Internet of Things will involve collaboration in various fields of knowledge. Analysis of IoT Protocols, various frameworks, challenges of IoT and Machine learning technologies are covered in the five chapters.

Due to resource constraints, heterogeneity, vast real-time data created by IoT devices, and the network's extensively dynamic behavior, optimization may not be possible. Machine learning and deep learning technique, which is the subset of ML, are applied to overcome varying obstacles. Researchers investigate how ML can address networking issues, such as device identification, security, traffic profiling, and industrial applications efficiency in IoT. This report has attempted to summarize the ML techniques for most critical implementations of IoT problems, including Deep Learning approaches and how they influence software testing.

TABLE OF CONTENTS

List of Figures

List of Tables

ACRONYMS

ACS	Auto Configuration Server
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
CoAP	Constrained Application Protocol
CPE	Customer-Premises Equipment
CSMA/CA	Carrier-sense multiple access with collision avoidance
DDOS	Distributed Denial of Service
DL	Deep Learning
DTLS	Datagram Transport Layer Security
E2E testing	End to End testing
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
HIPPA	Health Insurance Portability and Accountability Act
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IEFT	Internet Engineering Task Force
ISM bands	Industrial, Scientific and Medical
MAC Layer	Media Access Control Layer
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
OMA-DM	Open Mobile Alliance - Device Management
OSI model	Open Systems Interconnection model
PAN	Personal Area Network
PHY Layer	Physical layer or layer 1
QoS	Quality of Service
RFID	Radio-frequency identification
SE	Software Engineering
TCP/IP	Transmission Control Protocol or Internet Protocol
TDMA	Time-division multiple access
UI	User Interface
UX	User Experience
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Networks

Chapter1

Machine Learning

People learn from their previous experiences, and machines follow human instructions, but machines taught by humans could learn from past data and act much faster. It is more than just learning; it requires comprehension and reasoning, and it is called Machine Learning(ML) (33). A branch of ML is closely related to statistics, which uses machines to make predictions, optimization techniques, and implementation to be delivered to the world of machine learning (34). Data mining is a similar area of research focused on exploratory data processing by unsupervised learning.

The ML will provide a potential benefit for computer networking. Some researchers have found that ML can solve networking problems, such as congestion, route selection, and security, considering ML as the primary technology for supervising autonomous smart/intelligent systems. However, controlling and maintaining IoT systems is difficult. ML helps analyze the data to extract knowledge from it by connecting multiple devices, to meet end-user needs. Devices are now increasingly intelligent, like an "embedded intelligence" (33) (35).

Most IoT systems are becoming complex, heterogeneous, and dynamic; Moreover, such system's services need to be enhanced in terms of usefulness and variety to attract more users (35). Much has been done to apply ML techniques to IoT. ML's application enables users to obtain deep insights and create productive, innovative IoT applications because ML can provide feasible solutions to mine the information and features concealed in IoT data.

This project is to survey the implementation of ML for IoT by summarizing the existing applications with inherited challenges of the present technology. As a general overview, the objectives of this project report are to:

- Evaluate the utility of ML in traffic profiling, IoT device identification and other fields. The benefits of different solutions, including deep learning techniques, are explained.
- Provide an overview of IoT system architecture by examining various approaches protocols.
- Evaluate the utility of ML, DL & IoT in Software Testing is also included in this work.

1.1 The upbringing of Machine Learning

The first manually operated computer machine, ENIAC, was invented in the 1940s. At the time, machines were used to help math and science, so ENIAC was considered a "computer." From the very beginning, it was premature to create a computer that could imitate the human mind in 1952; Arthur Samuel is credited for naming machine learning, which acts as a foundational concept in modern technology (36) (37). Algorithms and neural networks are used to support computers in performing more efficiently and effectively. ML Algorithms use data sets, or "training data," to automatically create mathematical models that make decisions without being specifically programmed (38).

Arthur Samuel of IBM invented the computer software for checkers in the 1950s and then used it to play interactive checkers (39). Since the software was designed to run in a minimal amount of RAM, Samuel performed alpha-beta pruning. His plan included a weighted scoring based on where the pieces are on the board. The scoring function attempted to estimate the probability of the outcome of the referendum. The program uses a minimax algorithm, which generated the minimax strategy (39). Eventually, it became the first computer program that could beat a checker's world champion. This program taught checker players strategies that help improve their play. Around the same time, Frank Rosenblatt developed an even simpler neural network classifier called the Perceptron.

The neural network area was stagnant for several years due to its failure to solve specific problems. The Mark I Perceptron was the first active neuro-computer. However, it demonstrated minimal capabilities, and the Perceptron did not correctly distinguish several different types of images was an issue with the study (39). Neural network research failed until the 1990s, but it took many years for investors and funding agencies to accept this reality. Statistics brought ML to prominence in the late 1990s. Statistics and computer science contributed to modern techniques in artificial intelligence research (40). As vast data becomes available, scientists have started developing artificial intelligence systems to analyze and learn from such data. IBM Deep Blue machine defeated chess grandmaster Garry Kasparov in the world championship (40). However, Deep Blue is not functional anymore, so this is a piece of history.

We may interpret the 90's as an age in which ML flourished. From the 1990s to the 2000s, significant advances occurred in the area. Not only were they improving the technology, but the hardware and processing speed were also improving. Over the past few years, computers have not only grown in power but also in scale. Some of the past noticeable achievements are shown in table1 (29).

1763	Thomas Bayes's works,An Essay towards solving a Problem in the Doctrine of Chances is published two years after his death, having been revised and amended by a friend of Bayes, Richard Price.
1805	Adrien-Marie Legendre defines the "méthode des moindres carrés" which can be found in English as the least squares form used for data fitting.
1957	Frank Rosenblatt invented the perceptron while working at the Cornell Aeronautical Laboratory that got public attention.
1992	Gerald Tesauro created TD-Gammon, a computer backgammon software that uses an artificial neural network trained with temporal-difference learning. Human's play at a similar stage of backgammon, but TD-Gammon does defeat top humans.
1997	IBM's Deep Blue beats the world champion at chess.
2009	ImageNet is established. The ImageNet database is a huge visual database imagined by Fei-Fei Li from Stanford University. Li found that the best machine learning algorithms would not perform well if the data did not represent the real world.
2011	IBM's Watson gathers and analyzes language to teach machines to do stuff like voice and translate languages. competition.
2012	The Google Brain team was able to build a neural network that learned to identify cats by watching YouTube videos.
2016	Google's AlphaGo program defeated the previous iteration of Go (computer program) beating humans (humans playing against a computer program).

Table 1: Comprehensive history of machine learning (29)

Recently, Stanford University described ML as the science of having computers act without being specifically programmed. ML is a field of AI that contributes to some of the most significant technological developments. It spawned many diverse applied ideas, including new algorithms for robotics, the Internet of Things, analytics software, and chatbots. ML models can learn increasingly complex models the longer they run, which helps improve their accuracy. In combination with emerging computational technologies, ML algorithms produce quicker and more effective performance. Combining analytics with ML will address a broad array of complexities within an enterprise. Modern ML models can be used to forecast everything ranging from the spread of disease to stock market volatility (37) (41).

1.2 Machine Learning

Data is not random, and it contains a structure capable of predicting or acquiring information like recommendations in E-commerce and OTT platforms. ML is a set of tools for computers that use data to optimize the system, enabling us to provide examples of how computers will perform tasks. ML algorithms first train the model on vast quantities of data and then make a prediction or decision. According to Arthur Samuel, ML is a computer's ability to complete a task solely based on its knowledge. According to McKinsey, ML is the algorithm capable of learning from data without using programming laws (42).

For instance, assume that we want to develop a program to differentiate between legitimate emails and unwanted spam. We might try to write some basic rules, such as messages with specific characteristics (such as the word "won lottery" or fake headers). However, it can be complicated to write rules to decide which text is legitimate or spam messages. Spammers deliberately alter their spam writing techniques (e.g., "L0tt0") to trick. It is an insurmountable task to write meaningful rules and keep them up to date. Luckily, ML was a solution. Examples are "learned," which means that we have manually marked some emails as "spam" (unwanted email) to the learning algorithm, and algorithms automatically differentiate them from each other. Modern spam filters are learned from examples. ML is a diverse and exciting area, and it can be described in three ways (43).

I. **Artificial Intelligence View**: For humans, learning is a core aspect of information, and intelligence is critical for creating intelligent machines. Years of development in AI have shown that creating smart computers cannot render all programming rules; automated learning is necessary. For example, we humans were not born with essential skills or abilities; whatever we humans execute is done by deciphering a specific object's data. Similarly, the machines can be made to function in the same manner.
II. **Software Engineering View**: ML helps to program the computers using an example, which can be much more comfortable than writing code in the traditional way.
III. **Statical View**: ML is the combination of computer science and statistics, where algorithms in computer science are employed on statistical problems. ML has applied to many contexts, and the issues it seeks to solve are not restricted to the conventional statistics class.

People believed a specific meaning by using historical data, such as how much they would save by making an individual monthly payment. After looking through hundreds of advertisements on the Internet, including ads for new vehicles, used year-old cars, 2-year-old vehicles, we can see that the car price depends on its age and decreases $1,000 a year, but will not get lower than $10,000. It is termed regression in terms of ML.

The issue with this is that the prices will change based on various factors, such as production date, many different choices, technological conditions, seasonal demand spikes, and several other secret factors. We will feed the machine some data and give it enough time to find hidden patterns regarding price (42). This is the main difference between humans and machines, i.e., the machine cannot understand the new data without required data sets. There are different means of analyzing the data; some of these methods include (43) (1).

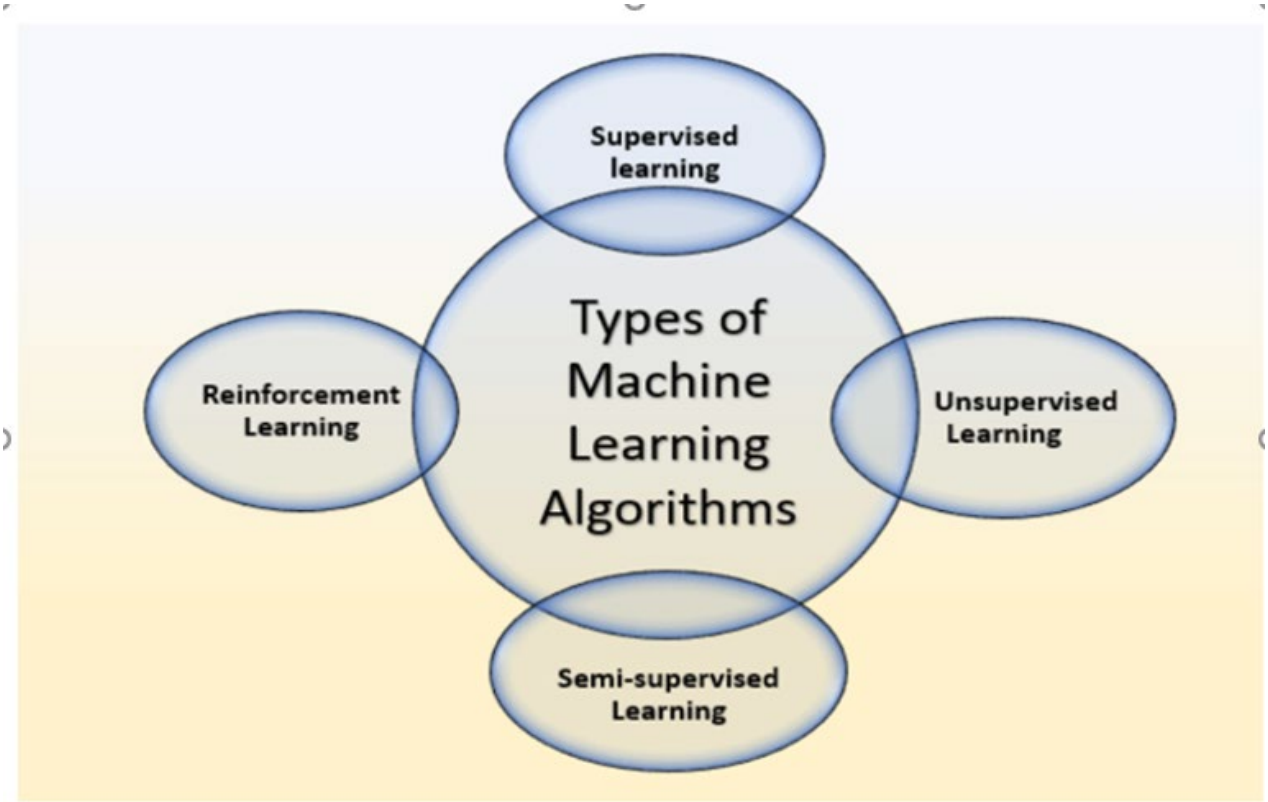

Figure 1: Types of Machine Learning Algorithms (1)

I. **Supervised Learning**: Training data is marked with the correct responses, e.g., spam, right. Both classification (where the results are discrete labels, as in spam filtering) and regression are two common types of supervised learning (where the outputs are real-valued). Algorithms such as Decision Tree, K-Nearest Neighbor's, Support Vector Machine comes under Supervised Learning.

II. **Unsupervised Learning**: We obtain a set of data in the form of individual observations, of which we must analyze and identify trends within. The two most significant examples are the concentration of data and grouping data. Algorithms such as K-means clustering, Mean-Shift, DBSCAN comes under Unsupervised Learning.

III. **Reinforcement learning**: An agent analyzes its act's effects to learn the best types of activities to take in the future. Algorithms such as Q-Learning, SARSA, DQN, A3C, Genetic algorithm come under Reinforcement learning.

IV. **Semi-Supervised Learning:** an approach to ML that allows the use of labeled and unlabeled data. Semi-supervised learning is the intermediate condition between unsupervised learning and supervised Learning in ML (with only labeled training data).

Two primary phases of machine learning (43):

1. Training model is a simplified version of the learning data based on the labels.

2.. Application to make a decision based on labelled data.

1.3 Benefits of Machine Learning

ML is a technology that has seen unprecedented growth in its use and popularity over the last couple of years. Some of the benefits are (2):

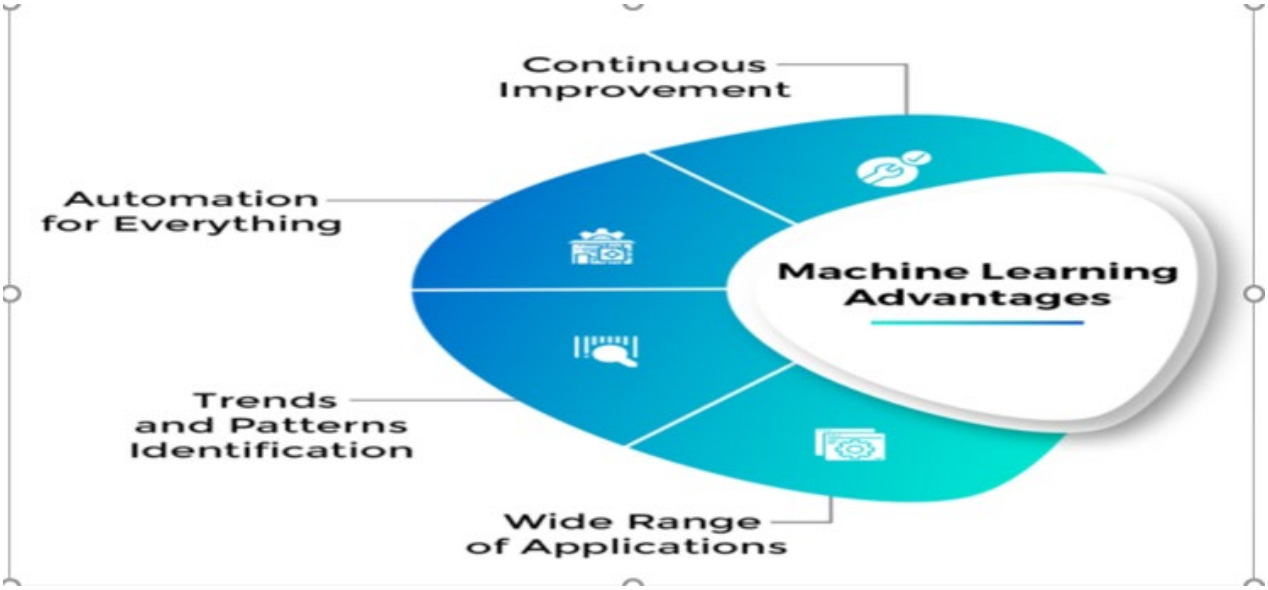

Figure 2: Benefits of Machine Learning (2)

- **Continuous Improvement**: ML algorithms use data to learn behavior patterns. If the prototype is checked further, the model's accuracy and efficiency increase. There is a vast abundance of data that is being continuously processed. The accuracy of finding associated items or recommending algorithms becomes improved by incorporating large quantities of online data.
- **Automation for everything**: ML can simplify all manner of decision-making. This helps developers to spend more time on more efficient tasks. In 2016, we can see how several social media firms use sentiment analysis and chatbots in real life. The moment a derogatory tweet is sent means an automated email will be sent to first-level support customers. ML is transforming society by automating its applications.
- **Trends and trends verification**: As ML practitioners understand, supervised learning algorithms, unsupervised learning algorithms and reinforcement learning algorithms can be applied to several problems. We examine such patterns with relentless amounts of data using this tool. Amazon analyzes its customer's purchasing histories and pre-orders to determine what the consumer will buy next.

- **Wide range of applications**: ML is used in different industries due to its impact on everything. Companies produce income, save money, automate, forecast future trends and patterns, and evaluate different trends and patterns from historical data. New applications are developed every day, including GPS monitoring for real-time traffic, text prediction, spell checking, and correction. ML is a new Artificial Intelligence branch where algorithms are used for computer vision, robotics, and other applications.

1.3.1 Machine Learning Applications in IoT:

Predictive capabilities are essential for the success of mechanical devices. ML calculations will "realize" what is commonplace for the machine and subsequently detect when something odd begins to occur by taking measurements from various sensors inside or on a computer. Companies are using ML to provide their computers with a proactive maintenance plan, reducing expensive downtime. Both Amazon and Netflix use advanced computers to figure out what they can do to benefit the consumer and provide recommendations for movies and TV shows.

In IoT, ML will significantly alter our privacy on a subconscious level. As is the case with the Nest Thermostat, it utilizes a computer device to look into our inclinations for warming and cooling and ensure that the house is at the right temperature when we return home from work or wake up in the morning (44). This document briefly covers the applications of ML and DL in Chapter 5.

1.4 Machine Learning in Software Testing

Not all ML theories have the same underlying principles, premises, and approaches, regardless of where they are developed, which software engineering methods are used for various techniques, such as standard statistics, heuristic learning algorithms, and neural networks. Additionally, software engineers must make resource allocation trade-offs because they must divide their limited resources between different tasks. Is enough data gathered and used by ML algorithms to assist software engineering decision-making? Testing is a priority of this section.

In evaluating software, various kinds of data may be gathered. Capture execution traces and coverage information for all the test cases. The value of failure data is found in both software testing and application testing. The completeness of test suites, the automation of test oracles, risk-driven testing, and localization of faults are all unanswered problems for implementing failed tests. The following figure 3 shows with an operation diagram the broad range of ML applications in software engineering and the sense of software testing. The process mentioned here performs execution of test cases, collects data, and provides the said data to a learner, which returns interesting "patterns," for example, failure conditions.

There is a standard pattern used in promoting or automating decision-making, such as finding faults in source code. Nevertheless, in many situations, we are yet to explore this. If the decision is made, it will produce a "plan," which may take one of several forms, such as a list of tests and inspections to be prioritized, an automated algorithm to identify faults (oracle), or future changes to a test specification.

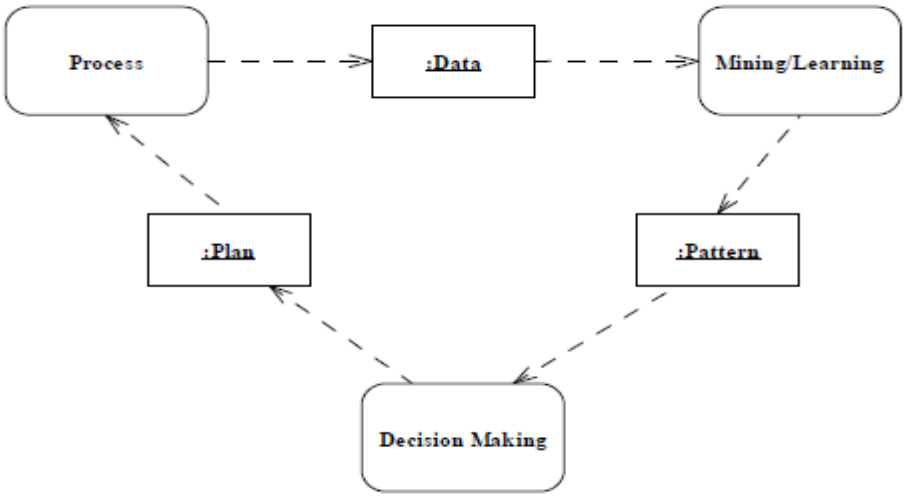

Figure 3: Role of ML in Software testing (3)

In Agile Projects, where requirements are not defined correctly and without a correct scope, defining test suits is challenging, especially when automation comes into the picture. Some software testing problems like a refinement of testing documents, debugging, risk-driven test execution and test oracles are proposed with ML techniques are mentioned in Table 2 (3).

Testing Problem	ML Solutions
Test specification and test suite Refinement	MELBA methodology.
Debugging / Fault localization	RUBAR methodology.
Risk-driven testing	Decision trees and neural networks algorithms
Test oracles	Segmentation algorithms

Table 2: ML application in Software testing (3)

Increased attention to ML models helps to support decision-making. It is crucial to decide the function of human input information and model exploitation (3).

- **MELBA Methodology:** In this methodology, the CP (Category Partition) technique seeks to create test cases covering various execution conditions. Test suites are translated into abstract test suites to help students excel. Abstract test cases include a mapping between inputs and a parameter that requires execution configuration instead of only the raw inputs (45). We use the test specification to increase the test case's abstraction level to learn more useful and exciting patterns. ML algorithm is used to learn rules that bind group choice pairs if a method for testing an algorithm has been introduced. Obsolete test cases and potential

new test cases need to be discovered; categories may be absent or ill-defined, thus necessitating an improved CP specification. A modification accompanies any iteration of improving performance to test documents. since we have the decision trees algorithm for thousands of abstract test categories, it does not matter whether it is run multiple times. (3). If the ML algorithm's rules have been mastered, the loop ends.

- **RUBAR Methodology:** A fault that occurred during testing is commonly used for error localization. RUBAR integrates ideas similar to Tarantula (46). There are several faults, which means that test cases sometimes fail due to the exact cause. Failed cases of test results are thought to be faulty. Suspected faults that appear to occur in similar execution environments are identified. Decision trees are useful for debugging as they resemble experimental error conditions. Case studies have shown the failure conditions, as modelled by a decision tree, accurately predict failures. They can be used to debug as the tree's failure conditions accurately describe actual failure conditions (3). ML cannot classify which properties are essential when they are first discovered. In other words, the learning algorithm cannot identify precisely when tests fail. CP has given guidance and options to MELBA.
- **Decision tree and Neural Networks Algorithms:** ML decides which regression tests are required for each code update reducing the number of required regression tests. Facebook believes that ML can be used to catch faulty changes. They assume that they can only do this with a limited subset of tests, but it will only require running a third of all tests that transitively rely on changed code to detect faulty changes accurately. Computer programs in ML-driven testing have already outperformed humans. ML-driven automation yields quicker and much less expensive test automation than human-driven automation. An autonomous test suite yields faster and more effective implementations. It is a big boost for every VP of Engineering's budget. Automated systems can automatically build, manage, perform and interpret tests without manual intervention. Perhaps not all aspects of software development can be automated. However, automation of quality assurance testing is generally accepted within the industry because it has historically been conducted using human experience and the workforce. Many people claim that no computer will ever do the work of a person. Resistance to ML and double-down on human labor also place these businesses behind. ML-driven test automation is just getting underway today, but it is on the verge of becoming the industry norm.

- **Segmentation Algorithms:** ML techniques are used to test new images. During the initial learning process, experts must manually perform test segmentations. ML is used to train a model to distinguish consistent and inconsistent segmentations based on many similarity measures. A new version of the image segmentation algorithm currently under test will determine whether a further segmentation generated by that algorithm is correct or incorrect, based on whether it agrees with previous diagnostic classifications. This process removes the need for human involvement, making considerable savings possible during the re-testing of segmentations. New segmentations and judgments about their validity (as judged by an expert) can be used to construct a model using ML algorithms to predict diagnostic consistency. Other research is needed to discover the reasons for the model's remaining uncertainty and how to reduce it.

Depending on the testing stage, functionality or device components are most commonly tested for "risk." The danger is frequently characterized as a combination of the possibility of malfunctions and the severity of the damage. Several studies have attempted to devise models to predict the position of faults within files or classes. ML algorithms have become very successful because they often do not require stringent assumptions, such as logistic regression.

Some of them, such as decision trees and induction laws, are easier to understand and use for practitioners. While previous studies utilized structural software measures to predict faults, there has been a growing realization that other factors must obtain adequate prediction models. Although the input data varies, it typically includes technical complexity measures for components and recent release data for components and developer's details on their interactions with the system being modified. Once a fault prediction model has been developed, system components can be rated based on their probability of having a fault. Also, visualization is much simpler with trees (47).

1.5 Machine Learning Techniques

Over time, the focus shifted and adjusted to create possible and robust algorithms for computing systems. In several areas, such as bioinformatics, speech recognition, spam analysis, computer vision, fraud detection, classification, regression, including density estimation, ML methods have now been extended to use for several things in the last decade. In a system to render the IoT further effective and scalable, it is possible to incorporate learning algorithms through several computers, bearing in mind the topology and computer ecosystem, including its participatory IoT devices. IoT systems must gather information and transmit it with many other systems while maintaining separation from other computing systems. They ought to be aware of any issues based on context and learn from their gathered knowledge. The development of the word "Cognitive IoT" (CIoT) contributed to this need.

Support Vector Machines (SVM)

SVM is an ML algorithm that uses labeled training samples to identify data points. Essentially, the challenge is to differentiate those nodes into two sections. These sections are divided by boundaries that are as strong as possible (i.e., separation holes). However, according to the position individuals sit in, the fresh interpretation would be graded. An SVM algorithm that involves computing a quadratic function through linear constraints presents an exciting solution to the multi-layer neural network's non-convex and unrestricted optimization problem (4). SVM will allow us to identify nodes, acquire information, and conduct analytics appropriately in the extensive array or pool of IoT computers.

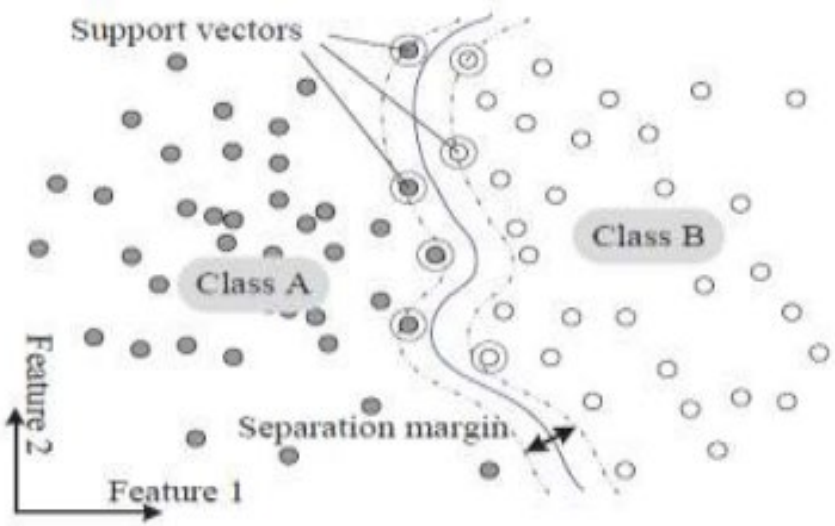

Figure 4: Example of non-linear support vector machines (4)

Artificial Neural Networks

Neural networking has sometimes been named this sort of learning method. It is fundamentally a learning algorithm based on the structure of neural systems and their relevant application. Given an interacting group comprised of artificially programmed neurons, different simulations become ultimately organized. ANN enables the use of the connectionist technique to evaluate the results. All the numerous modern neural networks become non-linear mathematical methods that are used in the simulation of information. They have traditionally been utilized to conceive the complex relationships that exist due to input and output information to find the connections.

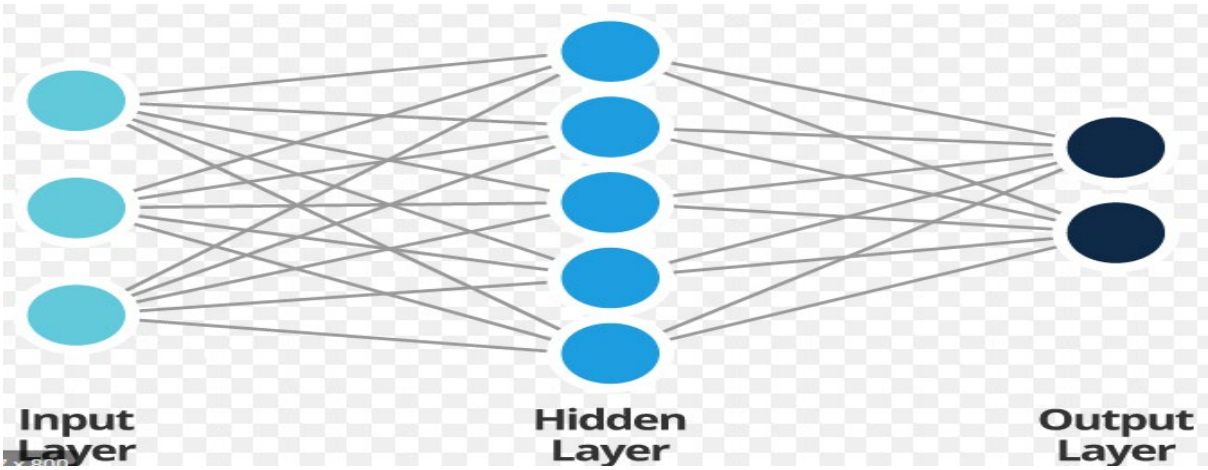

Figure 5: ANN algorithm (5)

K-means algorithms

K-Means solves node clustering, which is commonly utilized while it has linear complexity and straightforward implementation. In order to address multiple node clustering concerns, the K-means algorithm will be used. It arbitrarily selects K nodes for this with the various clusters to have been early centroids. It marks several of the nodes with a distance measure, including its nearest centroid and, using existing network subscriptions, it re-computes the clusters (6). It ceases if it notices that the convergence requirement is valid; otherwise, it loops back to the previous stage.

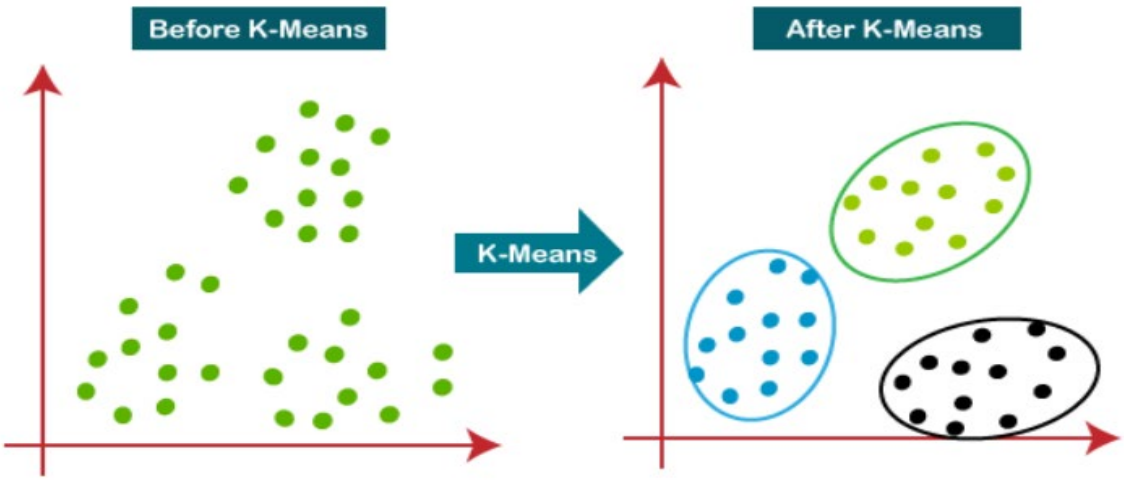

Figure 6: Working of the K-means Clustering Algorithm (6)

The k-means clustering algorithm mainly carries out two tasks (6):

- Determines the best value for K centroids by using an iterative process.
- Assigns each piece of data to its nearest k-center, and a cluster is created by the data points near the unique k-center.

Inductive logic programming (ILP)

It is a process to perform a reflection of multiple input instances, context information, and assumptions to control supervised learning only with logical programming. A hypothesized logical program can indeed be developed that uses an ILP scheme because we have to encrypt some general context information through unique collections of instances that provide a logical database,

including its details (48). Therefore, it includes all the positive and no negative occurrences that allow every programming language to reflect the theories.

KNN algorithm

It is among the simplest Supervised Learning-based ML algorithms. The KNN algorithm assumes the similarities between the existing and new cases and places them in a similar category. The KNN algorithm stores all available data points based on similarity. It can be classified into a proper suite category by using the KNN algorithm when new data appears. For regression, the KNN algorithm can be used simultaneously for classification, but primarily for classification problems. The KNN algorithm is a non-parametric algorithm, which means that the underlying data does not make any assumptions. It is also known as an algorithm for lazy learners because it does not learn from the training data. Instead, it stores the data set and then, at the classification time, acts on the data set (7). The KNN algorithm is given a new input data set, and it is kept in a category that is very similar to the new data.

A basic algorithm that holds all current cases and classifies new cases/data focused on a particular calculation is K Nearest Neighbors (e.g., distance functions). As an appropriate solution mode and has been used commonly for centuries. T implies the information seems to be in the domain of a function. KNN is a supervised learning algorithm that, primarily based on the identifiers (also called output values) of the neighboring feature vectors, qualifies a time series (further named a query point). This process essentially identifies the K types of clusters. The range within is negligible and more effectively identifying the category in which the new input data point will fall into, as shown in figure 7.

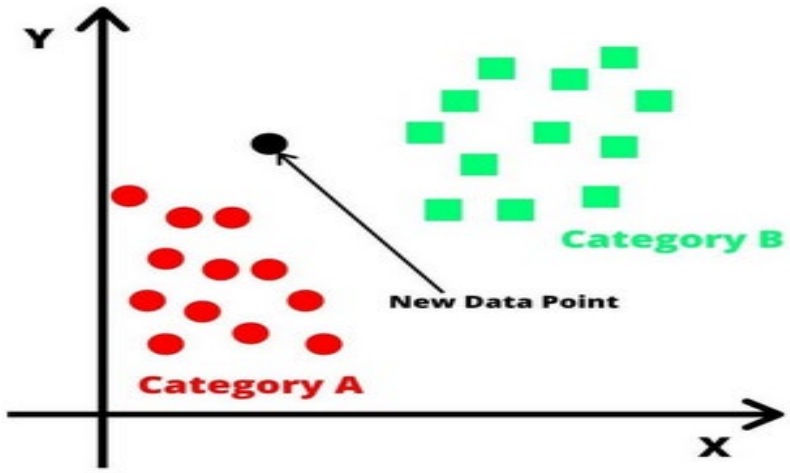

Figure 7: KNN algorithm (7)

The KNN algorithm's advantages are that it is more efficient when the training data is enormous; the implementation is very straightforward; The noisy training information data act robustly.

Bayesian networks

This graphical probabilistic strategy integrates a sequence of frames to describe a set of random variables, including all their subjective, makes it possible to compare. For example, it reflects the probabilistic relationships that occur among different seasons and inevitable consequences. If we know the signs, the likelihood of different disorders will accurately be determined by this network.

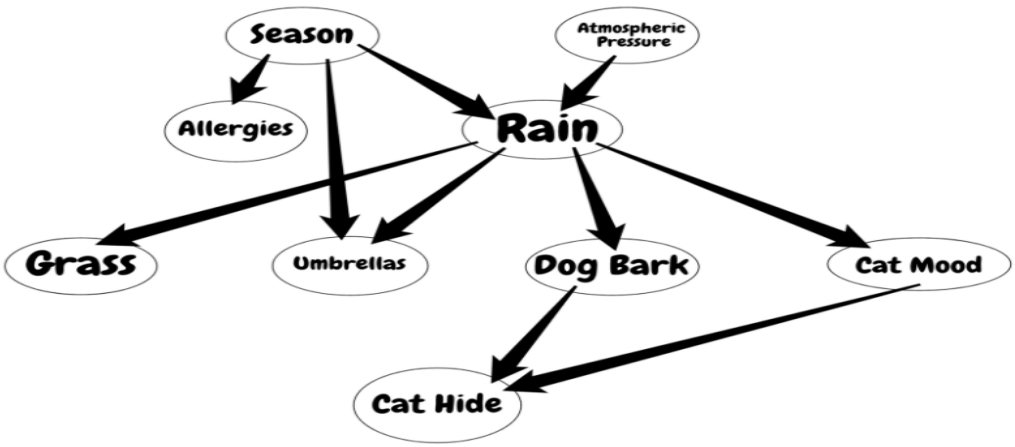

Figure 8: Example Bayesian Belief Network Representation (8)

Bayesian Networks are practical instruments for understanding the structure of the relationship of causality between variables. Once the model was already engineered, it can tell different things even with a small data set (49).

Decision Tree

A general predictive modeling technique with implementations covering many different fields is Decision Tree Analysis. Decision trees are typically constructed by an algorithmic method that, based on different conditions, defines ways to partition a data set. It is among the most frequently used technique, and it is practical for supervised learning. Decision Trees are a form of supervised and non-parametrical learning for classification and regression (50). The aim is to construct a model that predicts a target variable's value by learning basic judgment rules extracted from the data's characteristics. Generally, the rules of decision are in the form of statements that are if-then-else. The deeper the tree, the complex the laws are, and the model is correct.

A tree-like decision tree is a graph of nodes that reflect an attribute set and query; the boundaries display the answers (9). For the performance or class, the leaves represent the individual label. They are being used as a predominantly linear decision area in non-linear decision-making. Decision trees categorize examples by sorting them, using the leaf node classification from the root tree to a specific leaf node. For any specific element, each node of the tree serves as a test case, and each edge of that node is one of the possible responses to the test case. This method is recursive and replicated with every sub-tree rooted in the new nodes.

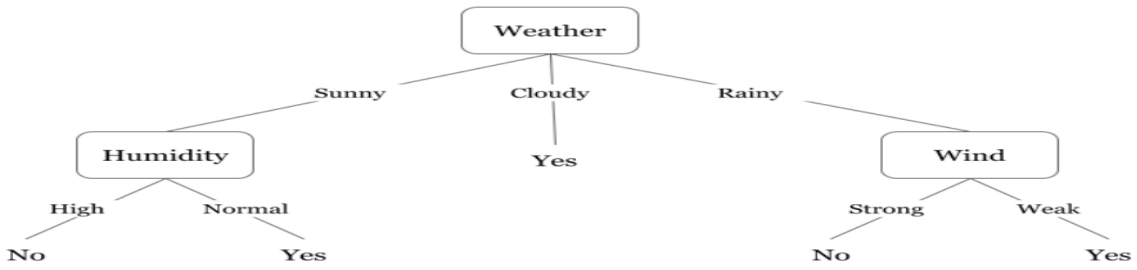

Figure 9: A decision tree idea for playing outside (9)

Chapter2

Deep Learning

Deep Learning introduces deeper neural network models, which could also derive more specific hidden characteristics and describe more intricate problems. DL seems to have more efficient functionality to generalize the complicated relationship of considerable raw information from different IoT implementations, unlike conventional accessible learning methodologies. The extensive and indispensable information resource will ultimately capitalize on DL (51). DL performance largely depends on the complex learning model's depth, coupled with fully convolutional architectures; therefore, it handles large quantities of data. Learning models should only be over-fitted to cope with the information flood. DL is something of an end-to-end learning system that can automatically learn how and where to derive proper functionality directly from the necessary information without any of the time-consuming and laborious requirements of handcrafted and manual functions.

2.1 The upbringing of Deep Learning

The planet is now witnessing a global AI transition that is impacting all industries. Also, Deep Learning is one of the primary motivating variables behind this AI revolution. Because of the help from giants like Google and Facebook, Deep Learning has become widely known, and it may seem as though it has only recently been discovered. However, one should be surprised to learn that DL has a history that dates back to the 1940s (30).

DL did not appear from anywhere; instead, it has evolved gradually and successively from the previous seven decades. A comprehensive history of Deep Learning history to recall the researcher's fundamental discoveries and how all these tiny baby steps played a role in kickstarting the DL revolution in our modern era (30).

1943	In their paper, "A Logical Calculus of the Ideas Immanent in Nervous Activity," Walter Pitts and Warren McCulloch illustrate the mathematical model of biological neuron. This McCulloch-Pitts Neuron has a very limited capability and does not feature any learning algorithm. While doing so, it will lay the groundwork for artificial neural network and deep learning.
1957	In his paper "The Perceptron: A Perceiving and Recognizing Automaton", Rosenblatt demonstrates a new iteration of the McCulloch-Pitts neuron — called the Perceptron — which was capable of self-learning binary classification on its own. This serves as an impetus for the subsequent advances in shallow neural network research for years to come, and until the coming AI winter.
1962	Stuart Dreyfus shows in his paper, "The Numerical Solution of Variational Problems," a simple derivative chain rule-based backpropagation model rather than a dynamic programming model that preceded it. This is yet another small step that helps deep learning for the future.
1980	Kunihiko Fukushima proposes the first convolutional neural network (which we now know as Neocognitron) that could recognize handwritten characters.
1982	The concept of the Hopfield Network, as conceived by John Hopfield, is nothing but a recurrent neural network. This serves as a content-addressable memory system, and will be critical for the development of future RNN models of the modern deep learning era.
1985	David H. Ackley, Geoffrey Hinton, and Terrence Sejnowski designed a stochastic recurrent neural network named Boltzmann Machine. This neural network has just an input layer and a hidden layer, but it does not have an output layer.
1986	Terry Sejnowski creates NeTalk, a neural network which learns to pronounce written English text by being shown text as input and matching phonetic transcriptions for comparison.
1997	"Long Short-Term Memory" is studied by Sepp Hochreiter and Jürgen Schmidhuber in paper "Long Short-Term Memory" (LSTM). It is a particular kind of recurrent neural network architecture that will, in time, revolutionize deep learning.
2006	Geoffrey Hinton, Ruslan Salakhutdinov, Osindero and Teh publish a paper where they stacked multiple RBMs together in layers and referred to them as Deep Belief Networks. For a large amount of data, the training process is far more efficient.
2008	NG's Stanford group began to advocate for the use of GPUs to speed up training times by many folds. This could lead to practicality in the realm of Deep Learning with regard to training on a huge volume of data.
2009	The Deep Learning community has always had difficulty finding enough labeled data. In 2009, Stanford professor Fei-Fei Li launches ImageNet, a database of 14 million labeled images. It would serve as a reference for all of the researchers in the field of deep learning who will take part in the ILSVRC competitions each year.
2011	In their paper "Deep Sparse Rectifier Neural Networks," Yoshua Bengio, Antoine Bordes, and Xavier Glorot demonstrate that ReLU activation function prevents the gradient vanishing problem. It means that deep learning experts will now have another way to avoid longer and impractical training times for deep neural networks.
2012	AlexNet, a GPU-based CNN model designed by Alex Krizhevsky, correctly classifies images in Imagenet with a winning accuracy of 84 percent. A 75% accuracy jump occurs. This victory sparks a worldwide deep learning revolution.
2014	GAN is also known as the Generative Adversarial Neural Network. With GANs, new doors of application in fashion, art, and science are opened due to their ability to synthesize real-life data.
2016	The deep reinforcement learning model developed by Deepmind has taken the human world champion in the complex game of Go down. This makes the game much more complicated than chess, so this achievement captivates everyone and takes the promise of machine learning to a whole new level.
2019	Yoshua Bengio, Geoffrey Hinton, and Yann LeCun are this year's recipients of the Turing Award for their outstanding work in the field of deep learning and artificial intelligence. This is a landmark moment for those who devoted their lives to neural networks in the 1970s when the

Table 3: Comprehensive history of Deep Learning (30)

2.2 Deep Learning

DL algorithms are merely sophisticated and numerically complex evolution of ML algorithms. Even with a practical purpose, the subject has attracted much attention: new observations are also related to findings that are not considered feasible. DL can likely make assumptions using a logic structure, as algorithms learn by observing and memorizing examples. Be aware that both supervised and unsupervised training will yield this result. DL systems have used a layered system with algorithms named an artificial neural network (ANN). It includes learning across structures that make a machine through simplified principles to build a hierarchy of sophisticated techniques. In DL, a system learns explicitly through text, sound, or photographs to handle problems and gain remarkable precision, much more than success at the more profound level (52) (53).

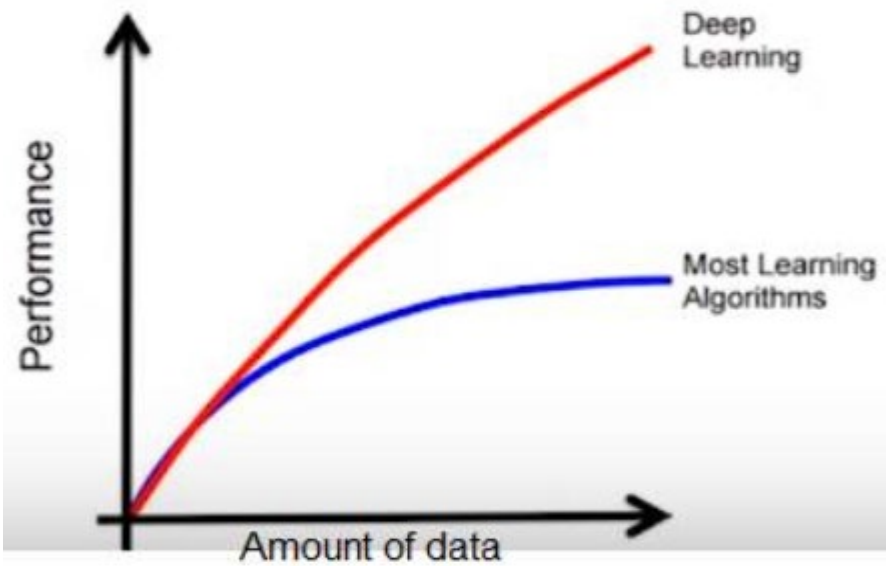

Figure 10: Deep Learning Efficiency (10)

DL algorithms identified as the coevolutionary neural network (CNN) perform the extraction and classification of characteristics. While also performs identification depending on various objects, CNN keeps track of feature extraction. DL algorithms are becoming more efficient as the amounts of data available increase. Also, as the volumes of information rise, several other learning algorithm's efficiencies drop (10). A CNN is another type of ANN, but it contains an input, an output, and some hidden layers. A CNN architecture may use a combination of pooling, convolutional, non-linear, subsampling, or FC layers for the hidden layers. Characteristics are trained again from input data in coevolutionary layers. CNN had learning parameters such as biases and weights between its elements(63).

Most DL techniques use neural network architectures, so DL methods are generally acknowledged as deep neural networks. Two critical stages of a DL experience are training as well as inferencing. The training phase could be seen as a procedure by which large volumes of data are labeled and their corresponding features identified. The program analyses these labelled features but recites them whenever needed to reach reasonable conclusions. The method makes assumptions and labels unexposed information, mainly during the inferencing stage, with the support of its acquired understanding. A significant feature of CNN is that different neurons even use the filters. Neurons represent components that add a transfer function to the number of their input signals, which is weighted. Non-linear layers are accompanied by convolutional layers, which transform all negative signs to zeros (54).

2.3 Deep Learning vs Machine Learning

ML includes a set of techniques to interpret and evaluate information, evolve from it, and make the appropriate conclusions learned from previous collections. On the other hand, to build an "artificial neural network," DL constructs the algorithms into several layers. This neural network could begin to make responsible decisions itself from the information (11).

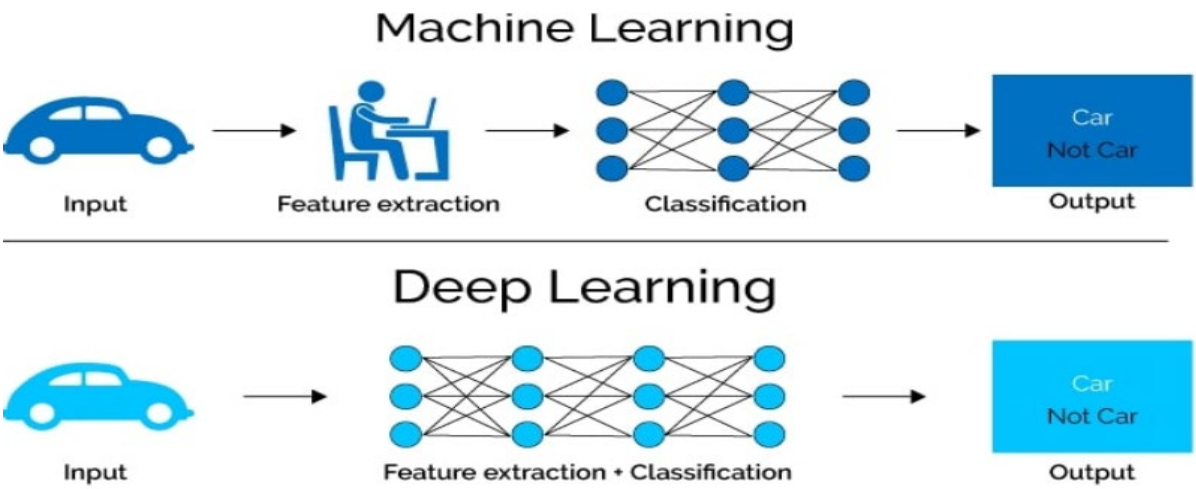

Figure 11: ML and DL comparison (11)

Dependencies on data

When the information structure expands, the much more notable difference between DL and standard ML has been its efficiency. DL algorithms do not do very well because DL algorithms need a large amount of data to comprehend them. On the other side, standard ML algorithms do not work well with a high amount of data throughout this case.

Period of Execution

Typically, preparation for a DL algorithm takes a long time. A DL algorithm does have many variables that involve longer than the average to develop it. It takes several weeks for the state-of-the-art DL algorithm ResNet to train things from scratch. Simultaneously, ML involves even less chance to practice, from several seconds to a few hours, in contrast. The DL algorithm takes even less time to drive during test time. Equate it with k-nearest neighbors, and testing time rises to maximize the model complexity. However, this does not extend to all ML algorithms because some still have short test times.

Problem Solving

ML algorithm is usually advised to cut down the situation into various sections whenever addressing the problem, tackle them separately, and merge everything to answer. In comparison, DL supports solving the challenge correctly; DL needs even more information than just a conventional ML algorithm. ML operates with a hundred data points, sometimes only with millions of DL points. A DL system requires a broad dataset to remove variations and provide high-quality interpretations because of the dynamic multi-layer structure.

Approach

ML algorithms aim to break information into components; then, all these components were merged to generate a conclusion or answer. DL algorithms had end to end process to solve a problem. They would also move via the following stages for machine learning: first identifying objects and afterward recognizing objects. On the other side, for the DL system, one must upload the picture, also with preparation, the system would revert through one consequence all the recognized objects and the position in the image.

Human intervention

A DL system gathers information on some characteristics without external human involvement for ML systems, which would need to define and shoulder the implemented functionality depending on the data kind. Most extensively, the machine trains on its own, the greater likelihood of an appropriate response.

Deep Learning	Machine Learning
A neural network is used to view the data and interaction's characteristics by moving data across computing layers.	Uses different types of advanced algorithms which change to model features and forecast data for potential intervention.
The output could be of any context, including free form features, including text document and audio.	For classification as well as scoring purposes, the performance is in numeric data.
In layers, DL architectures models to build an "artificial neural network" that could develop and take intelligent decisions by itself.	ML uses information parsing algorithms, learning from the knowledge, and creating intelligent decisions depending on what has already been discovered.
If they are put into development, algorithms became largely self-depicted in information processing.	Data analysts identify different techniques to analyze different methodologies in data sets.
Uses different automatic algorithms that transform to model features and causing the neural network to view data characteristics and interactions by moving data across computing layers	Uses different types of automatic algorithms that transform to model features and collect data for potential intervention.

2.4 Benefits of Deep Learning

Data on time series:

Unlike traditional data, the IoT data does have information on both feature vectors, timeframe. The traditional data analytical paradigms are not sufficient for the study of information of this kind. To analyze time-series results, DL models are indeed excellent. The RNN, as well as the LSTM, will quite well process information from time series. For this reason, these two architectures were designed explicitly.

Huge real-time information:

A vast range of highly deployed sensors dynamic IoT technologies is increasingly developing massive real-time environments. Broadcasting information within the exact defined time-lapse needs to be evaluated (55).

Heterogeneity:

Heterogeneous and distributed systems information produced through detecting structures is incredibly complex throughout the IoT. Moreover, it is a complicated task to interpret such complex information in various types, which reaches further than the capacity of traditional data analytical approaches (55).

Maximum utilization of unstructured data:

Some research indicates that a significant proportion of information along with the industry is unstructured. Much of it occurs in multiple formats, such as photographs, documents. It is hard to comprehend unstructured data with most ML algorithms that ensure that it remains unused, and this is precisely under which DL is more effective. To train DL algorithms, users could use different information structures but somehow provide insight that is important to the model training. For example, to forecast a specific company's stock market volatility, one could use DL algorithms and discover several emerging business analysis interactions (56).

Efficiency to have high-quality outcomes:

When effectively fully trained, similar to what one would mean for a human being, a DL algorithm could execute millions of regular, repetitive works within the same significantly smaller duration of time (56). Furthermore, only if the training data includes original information that still

does not reflect the issues people are attempting to address, the training's consistency in time is not the same in every case.

Removal about the need for identification of data:

Data labeling could be a time-consuming as well as costly task. When systems learn without even having been given any instructions, labeling is no longer required for a DL method**.** Such kinds of ML techniques were not quite as helpful as this kind of learning (56).

Eliminating the need for the engineering of features:

Feature engineering is a real job in ML since it enhances precision, and therefore, the mechanism can include spatial information over a specific topic. The necessity to carry out feature engineering on its own has been one of the key benefits of using a DL technique (56). An algorithm scans the information throughout this method to find features that overlap but then merge them to facilitate faster training despite directly becoming instructed. This expertise allows data scientists could save a substantial amount of time.

No Requirement for Data Marking:

One of the most significant ML problems is collecting great training information, although data marking could be repetitive and costly. Often, the method of data marking is straightforward, although time-consuming. When DL algorithms succeed at learning through standards, and a need for possibly the best system is measured outmoded with deep learning. For just this kind of learning, unsupervised ML is not sufficient (56).

Considering the above and considering the benefits of using the DL approach, it can be said that in the future, it is obvious to experience the effect of DL on various high-end technologies such as Advanced System Architecture or the Internet of Things (55). We would anticipate seeing even more important discoveries to the broader business sphere from linked and intelligent goods and services.

2.5 Deep Learning in Software Testing

The recent advancement in DL allows for the improvised handling of Software Engineering tasks. Industrial practitioners and academic researchers have utilized DL in SE tasks. Standard DL models and their variants are commonly used to solve SE problems. There are several concerns related to DL's applicability in practice, an issue that continues to garner attention. DL models, which are complex and challenging to understand, present a range of challenges, as mentioned below. Developing DL in software testing may be influenced by these factors. (57)

Effectiveness and Efficiency:

Recent studies show that using differential evolution to fine-tune SVMs achieves similar results to apply DL to link knowledge units in Stack Overflow (58). This method is 84 times faster than training DL models. As observed on code suggestion, an adapting n-gram language model explicitly designed for software surpasses RNN and LSTM. There is a trade-off between DL and lightweight domain-specific models like vars and classifiers with techniques like offline training and cloud computing. This sort of trade-off motivates DL research, such as investigating suitable datasets for DL and incorporating domain knowledge into DL.

Understandability:

Controlling DL understandability is a burden. Some recently proposed methods can help make DL more understandable. Facebook practitioners analyze to conceptualize large-scale deep neural networks (59). The proposed tool helps software engineers comprehend neuron activations, classifications, and DL activation patterns. Such a tool is a beautiful way to help beginning learners comprehend DL.

Testability:

The complexity of the model makes DL infeasible in SE. So, researchers strive to increase DL software testing predictability, such as DL testing (60). DL models are tested using coverage, and metamorphic testing Coverage testing is being used to test neural units in deep learning. Metamorphic testing yields test oracles. The findings show that SE techniques are critical for validating AI techniques like DL. To conclude, this hot topic is still an emerging one for SE practitioners and researchers.

2.6 Deep Learning Techniques

The Deep Neural Network (DNN) contains many computation layers capable of learning feature representation from input information. The mammalian brain's features mimic the functioning of the DNN. Multiple neuron processing units form the layer; A neuron calculates the weighted average of parameters. Indeed, the corresponding amount is transmitted as an entry to something like an activation function that generates the required outcome. The weight and bias have been set up for training and correlated with each neuron. The DL algorithm transmits the information across different layers; each layer can successfully remove features and transfer them to the next layer. Low-level attributes are derived from initial layers; then subsequent layers incorporate characteristics to form a complete model. DL can be broken into two scenarios, equivalent to conventional computer learning: Unsupervised Learning and Supervised learning. For a good outcome, hybrid learning blends supervised as well as unsupervised approaches.

Unsupervised Learning

Unlabelled data processing can indeed be worked out quickly. An unsupervised learning method should be used to deal with large unlabelled datasets. For secure initialization, back training can be performed with stacked RBMs or stacked autoencoders, range, as well as worldwide fine-tuning. Neural networks are trained by supervised learning utilizes named results. We have the data input and, therefore, no output to trace correspondingly. The purpose of this learning is to find information through modeling network connectivity. Algorithms can indeed identify the exciting structure representation of the data. Unsupervised learning uses clustering problems and correlation problems (61).

Boltzmann Machine (BM)

The BM is a network that is just a neuron-like structure that is randomly connected, liable for stochastically deciding whether to be active or inactive. BM addresses computational issues such as selection, Optimization, as well as learning difficulties. In the classification algorithm, several characteristics are discovered that illustrate very complicated actions in the testing dataset. In the learning algorithm, several characteristics are discovered that illustrate very complicated actions in the testing dataset. For classification as well as dimensional reduction, the Boltzmann machine is often used. There are two layers in the Restricted Boltzmann Machine (RBM): the

transparent and hidden layers (12). The transparent layer receives the data and the latent variables preserved in the hidden layer. Hardly any specific neurons inside the same layer are related, nevertheless. In both layers, the tendency is often related to any neuron. To maximize channel contact weights, the training phase includes back spreading and gradient descent strategies.

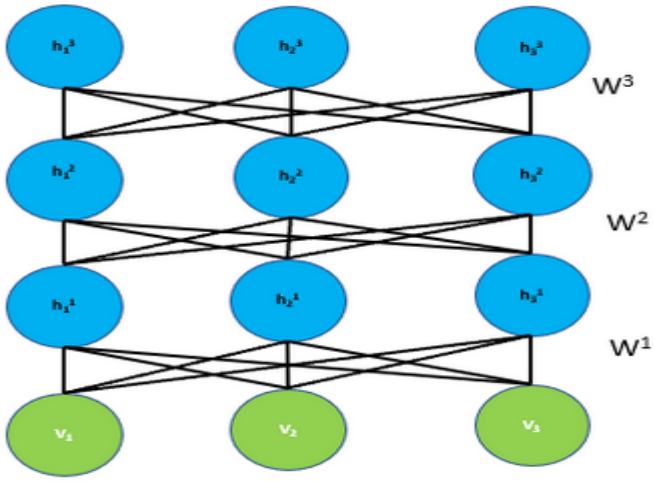

Figure 12: Graphical representation of Deep Boltzmann Machine (12)

Convolution Neural Networks

In the form of images or a speech signal, a Convolutional Neural Network (CNN) collects a two-dimensional input and pulls out hierarchical features through a hidden layer pattern. In the context of a CNN, the hidden layers consist of convolution layers and an attached layer. The convolution layer consists of filters of the very same structure as the input material. The buffer proportions are, nevertheless, smaller than the original dimensions. The convolution layer's results are the function maps often acquired mainly by input and filter input vector. It is forced to minimize the time of measurement and discourage over-fitting. The rectified linear units (ReLU) that involve neurons with softmax activation features are another significant part of CNN (62). There is indeed a particular role for each neuron of its subsequent layer, as it was only liable for maybe a portion of the input.

Recurring Neural Networks

RNN is a type of DL network utilized primarily whenever sequential data or series data, i.e., picture, voice, are available. In general, the RNN held the information from the existing state to a desired future. It is referred to as periodic that with each input, it executes this very same operation, while its output is dissimilar since it still relies on previous measurements. For modeling time series

assignments, Feed-forward Neural Network is not sufficient. T Recurrent Neural Network (RNN) has been created (63). The latest input test and the previous test are primarily supplied to the RNN as input. At period, the RNN performance relies mainly on RNN output at period step-1. The outcome of each neuron is provided as input to the corresponding stage for this reason. Every neuron in RNN is equipped with an internal neuron and store the preceding input material in the database. A variation of the Back Propagation algorithm, known as Back Propagation Over Time (BPTT), is used for training the network (13).

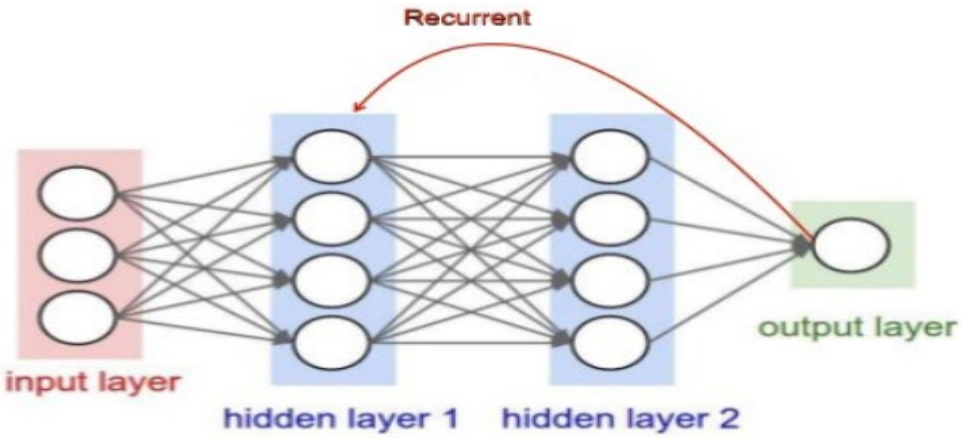

Figure 13: Recurrent Neural Network Layers (13)

Deep belief network (DBN)

DBN is a probabilistic algorithm for DL that is unsupervised. It has several levels of parameters that are secret. It takes more hidden layers to solve complex situations; each layer has a special mathematical relationship with most of the other layers. DBN should learn probabilistically; BDN requires supervisor preparation to carry out classification after training. In DBN, each layer is known as an RBM (or DBN), and the subsequent layer serves as an input layer for a different layer (14). The DBN should be used for cluster identification and develops information for images, video sequences, and motion capture.

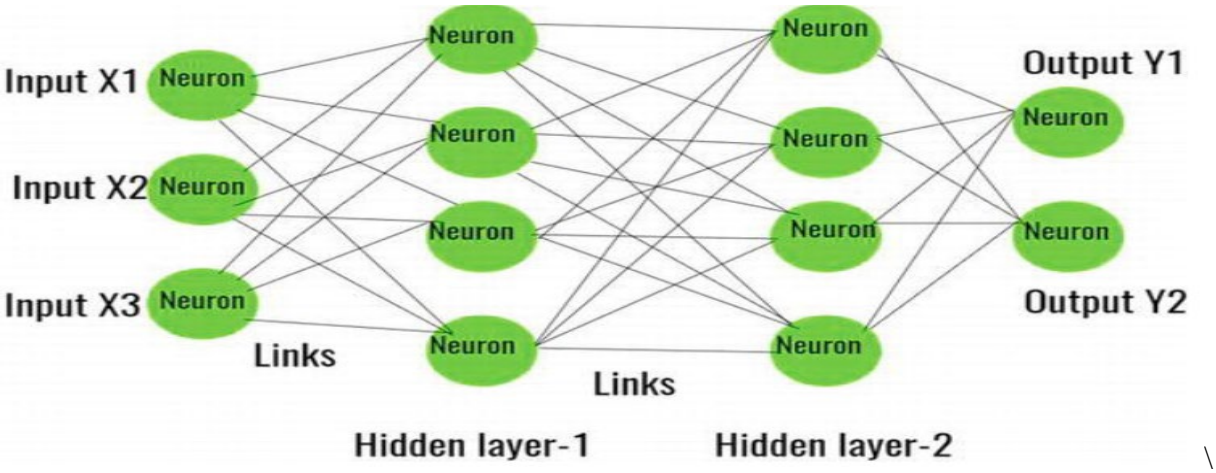

Figure 14: Deep Belief Network (14)

LSTM (Long Short-Term Memory)

When modeling a series of data issues with long-term dependencies, RNN suffers from the vanishing gradient problem. A new DNN architecture named Long Short-Term Memory has been developed to solve this constraint. The LSTM device consists of a memory cell, a written gate, a forgotten gate, and an output gate. The memory cell holds attributes, and the gating circuits monitor the flow of information in and out of the memory cell. Sigmoid or hyperbolic tangent elements have been used in the gates as activation functions. The gradient descent combined with the BPTT algorithm is used to train BSTM units in RNN (15).LSTM is used for unsupervised learning; recent applications are OpenAI and Starcarft2 games.

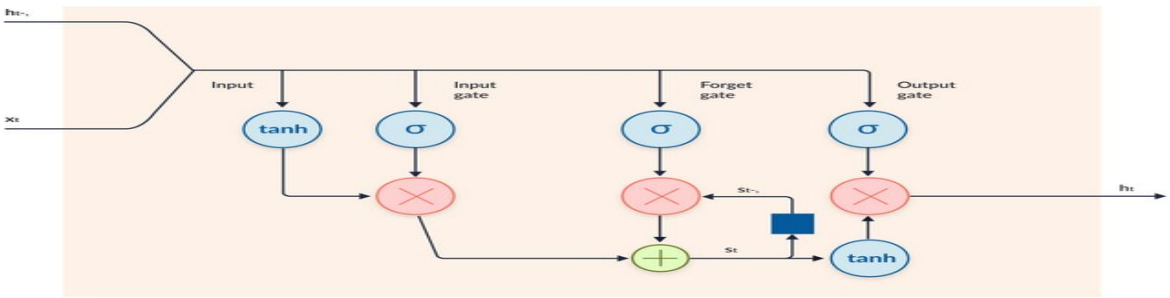

Figure 15: LSTM Network (15)

Chapter3

Introduction to the Internet of Things

"Internet of Things" is an interconnection and integration of the physical world and cyberspace, such as physical attributes vary from environmental conditions to heart rate, that is seamlessly incorporated into the information network and communicate data associated with user's wearable devices. Kevin Ashton coined the word "IoT," which described it as sensors connecting to the Internet and operating like the Internet by having transparent and ad hoc communications, openly exchanging data, and enabling computers can recognize their environments to become the human nervous system[50].

3.1 The upbringing of the Internet of Things

Franz Dill proposed connecting devices in 1970, a bar code idea to identify grocery stores. Later in 1982, at Carnegie Mellon University, Michael Kazar, a graduate student, connected the Coke machine to some terminal concentrator and modified source code, i.e., could check ten times a second for the numbers of coke bottles in the machine and notify when the device is empty. In the year 1992, graduate students at Cambridge University connected a camera to an internal network to monitor the coffee level in a coffee pot shared by multiple floors in seven store building. Later, Coffeepot Camera became a world-first webcam created to benefit other fellow people in the building who were disadvantaged to have a bad or left-over coffee (31) (32).

Kevin Ashton & his team at Procter & Gamble (P&G) tried to solve the existing barcode system in 1999. The issue is distinguishing similar products like regular milk from chocolate milk; it helps in accurate bookkeeping by linking new RFID(Radio-frequency identification) to the Internet. Kevin came up with the term IoT-Internet of things. Since about 2000, we have had convenient access to the Internet; in the same year, LG announces it is first Internet refrigerator plans. The critical publications of IoT, including Science American and the Guardian, mention IoT in 2004.

Furthermore, RFID is deployed into the commercial world. ITU(International Telecommunications Union) released the first study on the subject, referred to it as a completely new dynamic network of networks in 2005. The same year Aldebaran Robotics created a Wi-Fi-enabled rabbit, i.e., which gives notifications about new updates to end-users (31) (32).

Official IoT was born at the First European IoT conference in 2008. In the same year, over 50 companies Bosch, Cisco, Ericsson, Intel, SAP, Sun, Google, and Fujitsu, formed an IPSO

alliance to advocate for IP networked devices in energy, consumer, healthcare, and industrial applications. Later in the year 2011, it became a buzzword in the software industry as several companies promoting educational and marketing initiatives on the topic. In 2020, simple tools like AWS Deep lens were readily available in the market for students and developers to develop new IoT applications (31) (32).

1970	Franz Dill -proposed the original idea of connected devices
1982	Coke machine connected to the Internet
1990	Coffeepot Camera connected to the Internet
1999	Kevin Ashton was coining the term "Internet of Things."
2004	IoT deployed into the commercial world like Walmart
2005	ITU published the first report & Wi-Fi enabled rabbit developed by Aldebaran Robotics
2008	First European IoT conference & IPSO
2011	Awareness in the market increased

Table 3: Comprehensive history of IoT (31) (32)

3.2 The Internet of Things(IoT)

The main objective of IoT is to allow the best possible level of connectivity. On the Internet, items such as furniture, vehicles, phones and objects can interact with other objects and people. Additionally, they can send and receive information from one another. IoT is an emerging technology that helps physical devices to sense and control the world around them. It makes it easier to use and build modern and innovative applications. To achieve this, it needs to be some form of intelligence in devices that will join them to the network.

Different definitions of IoT exist, and it is a good idea to mention some of them (64):

- Gartner described IoT as a network of physical objects that contain embedded technology to communicate and interact with the inside or the outside.
- The ITU(International Telecommunications Union)describes IoT as: "A global computer network for smart devices, allowing advanced communication and data sharing between people and things. (like machines) based on current and evolving communications and information technology."
- Institute of Electrical and Electronics Engineers describes IoT as a network of objects equipped with an embedded sensor linked to the Internet.

- IEFT(Internet Engineering Task Force) defined IoT as "connecting," involving everything in the environment(electronic, electrical, non-electrical) to be involved in providing contextual information when searching for meaning within the "outside" context.

Based on the above definitions, IoT has the following characteristics:

- IoT is a promising technology, which includes linking and integrating objects.
- Objects have a unique identification, which makes them recognizable.
- Objects communicate with one another using the Internet.
- Objects can perceive and interpret their surroundings.

IoT allowed smart objects to see, hear, make decisions, perform tasks by providing contact with other objects, and exchange information. Utilizing the underlying technologies like RFID, universal and widespread computing, wireless sensor networks, embedded systems, internet protocols, IoT transforms old-fashioned analog devices into smart ones.

The smart objects are designed to be adapted to different technologies because the underlying technologies can be used regardless of the application they are being used (65). The systems of the Internet must be updated to take advantage of the IoT.

Also, IPv4 would not be adequate to accommodate the increasing number of smart objects connected to the network. Considering the security and privacy threats of an IoT environment, a new challenge to manage information sharing across devices. The management and control of IoT are essential to provide high-quality, reliable service delivery for consumers (65).

3.3 IoT network vs Traditional IT networks

The IoT design and structure are readily different from that of the "Enterprise IT" network, as both have been developed over the years to meet current network service needs. There are many similar things between the IoT and traditional IT networks, but the requirements, challenges, and management of IoT systems are different.

The IoT network relies on the Operational Technology (OT) network, which manages the operational systems. In contrast, traditional IT networks mainly focused on the infrastructure or data flow and did not care about the type of data. In goal-driven operations, data is the primary parameter that differentiates these two definitions.

The IoT framework design pays attention and the transportation of data, collection and analysis of information. In contrast, the IT network framework presumes the secure, constant support of business applications/projects. That difference between IoT and IT networks can be defined as such (18).

I. **Security:** Tools could be physically accessed externally but are also accessible using spectrum analyzer technology. Seeing that any IoT device in the outside world is revealed, any such device must be protected. Here, device-level authentication is recommended. Devices and the entities in the IoT network should all be registered, protected, and regularly monitored. The data exchanged by endpoint devices to back-end applications needs to be encrypted.

II. **Scalability:** The IoT network has a great degree of scalability and is not supported by IPv4 address spaces or routing techniques like NAT & PAT. It only supports IPv6 due to the increasing scale of the network from several thousands of nodes to millions of endpoints.

III. An IoT network can include numerous devices, like IP-enabled smart endpoints and non-IP-enabled devices like old manufacturing machines. Thus, there should be a method for translating API use scenarios. In IT networks, legacy devices are not considered troublesome and can be fixed by updating or replacing them.

IV. IoT devices have limited resources. IoT sensors are lightweight, cheap and can provide single-function capabilities. So they have no memory and low-powered CPUs to use. While IT networks, based on secure, efficient components, utilizes high-speed communication. The wireless technologies should function over long distances, which requires new protocols for the network and transport layers.

IoT provides data in real-time, then processes it to show information, and retains the findings. The data is unstructured and only contains general information, but the information extracted from this data may provide helpful business models and valuable information for potential use.

3.4 IoT Architecture

Different IoT architectures were introduced during the time and considered unique characteristics and features. Almost every model was built over multiple parallel domains or in multiple layers. Moreover, each layer works independently from another. Each of them performs specific tasks and functions and consists of certain protocols, but some protocols may run and reside at more than one layer. Each of these models aims to connect things or objects to the network and communicates with applications. The IoT models can be in parallel or vertical structures. This section discusses the simplified architecture for IoT, as shown in Figure 16.

IoT is divided into two separate stacks in the simplified architecture: IoT Data Management and IoT Core Stack. Moreover, they work in parallel, and almost every model has this Core IoT stack function (66).

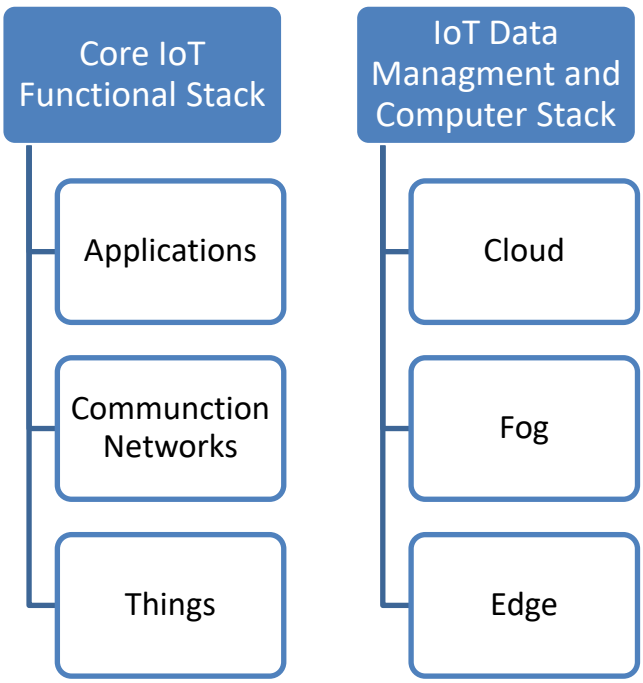

Figure 16: IoT Simplified Architecture

For studying IoT structures, we look at real-world artifacts. The smart object sensor communicates with the network and the application that manages the smart object through connections to the network, and this is called the Functional Stack for IoT. Inside the Stack, there are three layers.

At the bottom of the model, "Things," smart objects and sensors exist; sensors can connect with

wireless access points, produce data, relay data, and use the environment's constraints to match. In the next layer, network communication will occur in the WAP(Wireless Access Points) or Gateways. If smart objects need to connect with external sources such as computers, they may communicate wirelessly. So, smart objects will have a standard Gateway, and that Gateway will collect Smart Object data, and this data goes into one central database for processing. The core data location is generally at a data center or in the cloud. The Gateway is an internet-capable computer. It forwards information by packeting or relaying.

We will control the smart device's network at the Application layer and not the data layer. Applications can record device behavior based on the collected information, which is then processed via an algorithm.

The foundation of the IoT smart devices that perform specific tasks and functions. Some behavior may be performed without the need for external systems. However, smart objects maintain their stored information and share it with other devices or external sources in most situations. This management system is designed to act solely based on data collected (31). Core IoT Functional Stack is a hierarchy of three layers (67)

- **Perception or sensor Layer**: This layer uses sensors and radio frequency identification tags that can receive and transmit information from the environment and then forward the information to higher layers to be more processing.
- **Middleware Layer**: This layer allows for the fast transmission and reception of network messages, assignment of IPv6 addresses, and objects for communication and data collection and processing.
- **Application Layer**: Performs application-specific security and business management functionalities.

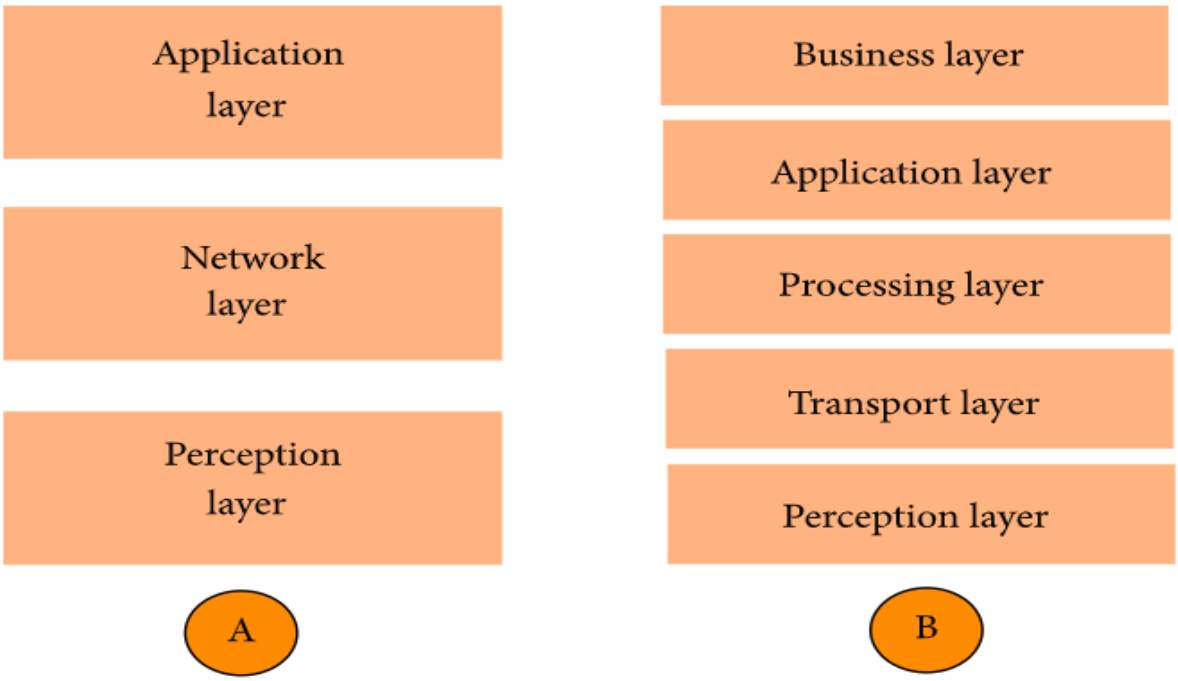

Figure 17: Architecture of IoT three layers and five-layer

With the Main IoT Usable Stack in place, a continuous stream of data is captured by sensors and delivered to the infrastructure. Data management is divided into its layer or feature intertwined with the other domain or layers (68). As shown in Figure 16, IoT system, cloud, and IoT platform, there are three layers for data management.

- Edge Layer: sensors will provide information and process with the data themselves. There may be "Mist Layer" or "Edge Layer," which handles the processing to avoid any problems. Data will be stored in the sensors.
- Fog Layer: As an intermediary, the sensor transfers some of the information to the network's central node. The central node of the network independently obtains other information from the sensor. A Gateway and the network manage the Security and Traffic Control system.
- Cloud Layer: Applications that are implemented in the cloud or central data center can handle the data.

3.4.1 Advantages and Disadvantages of Simplified Architecture

The key benefit of the architecture is its simplicity. Data processing is centralized and performed outside the smart cloud and then sent to a cloud provider. Given this, smart objects can link to cloud applications. Because the amount of data in IoT networks has increased, several problems include latency and a large packet size of the network backbone. This latency affects the network's performance and network bandwidth management of IoT networks, making the data analysis, storage and processing step very complicated (18).

3.5 IoT Components

As of now, the Internet of Things (IoT) is a personal transition of connecting our smart devices and objects to function and navigate remotely. Three simple components go into the Internet of Things, as shown in figure 18 (16).

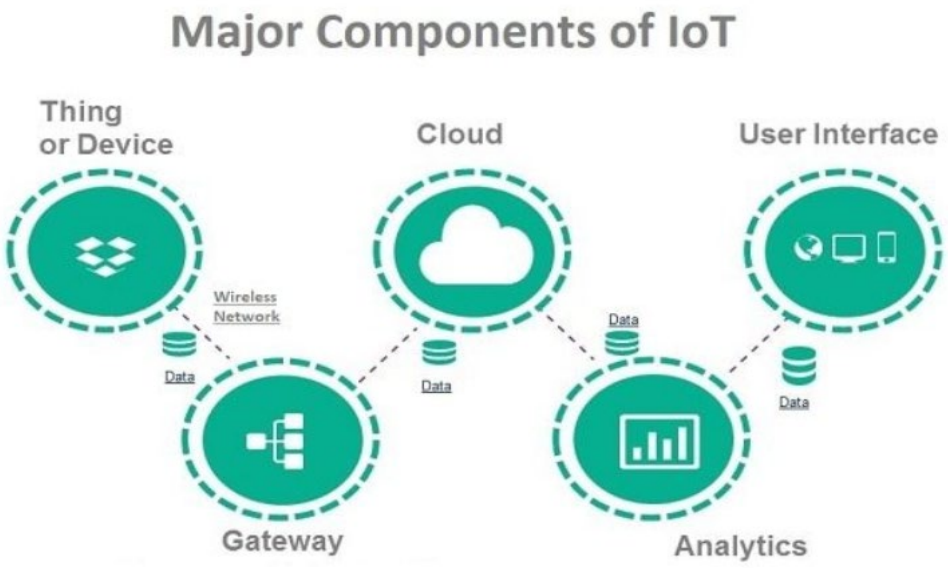

Figure 18: Major Components of IoT (16)

1.1.1 Smart Things or Objects

IoT networks include all artifacts or things that can be linked to the network, including legacy devices. This creative stuff was discussed as smart sensors, IoT devices, intelligent devices, and smart objects. Intelligent objects must have the following characteristics (16).

- Power: A smart object needs to manage its responsibilities and be available. Any type of energy source can drive the intelligent object.
- Sensors and actuators: An entity has a sensor to sense, feel, track the world, and have its action to take.
- Processing Unit: For processing, receiving data and analyzing information received by sensors.
- Communication device: To connect smart devices to a network or Gateway, we need a way to communicate, requiring either wires or wireless transmission/signals.

IoT objects can be divided into several categories, as shown in figure 19.

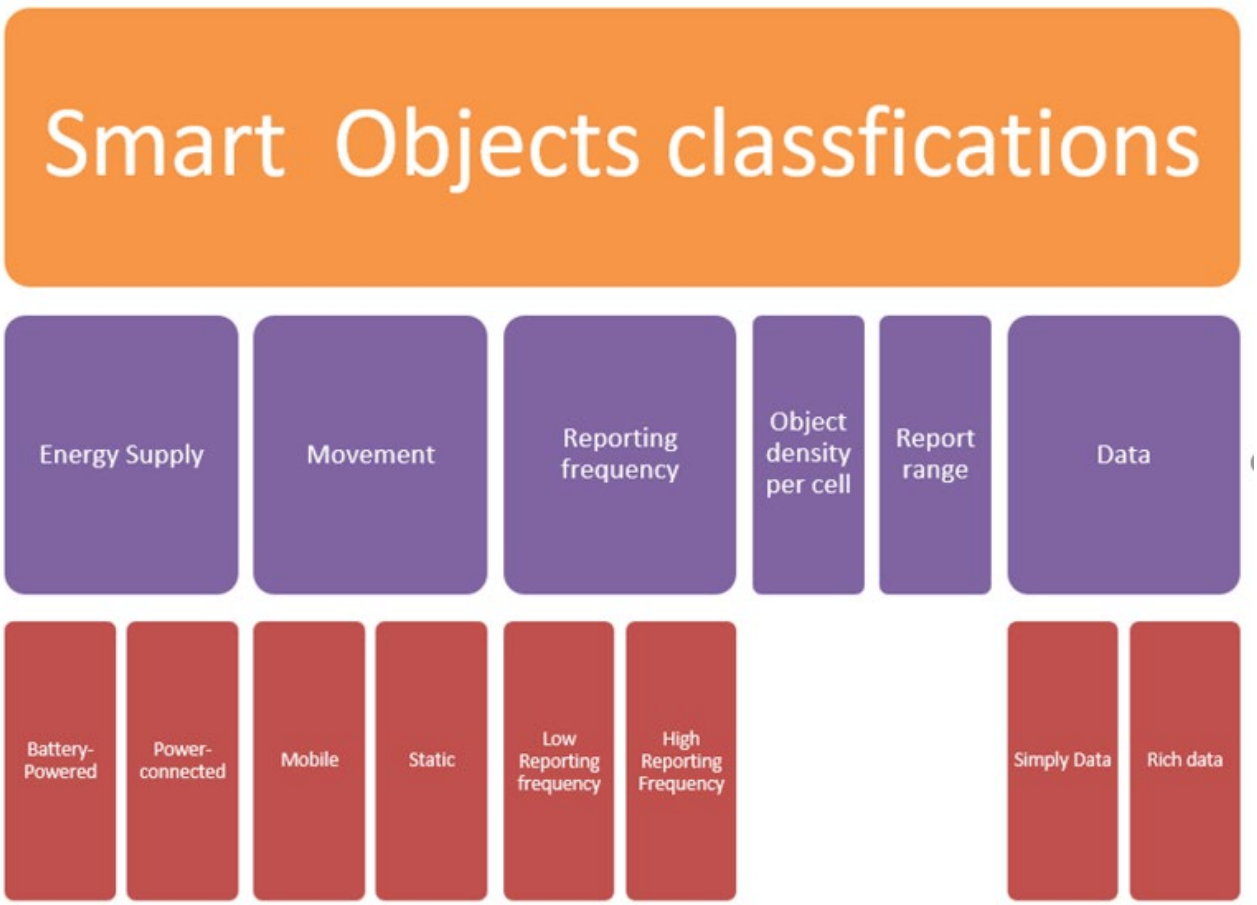

Figure 19: Smart Object Classifications

- Energy Supply: This category is based on whether the object is powered by an internal power source or is powered externally. The distinction is that Battery-driven objects are quicker than those powered by other means, but they are constrained in their travel speed and energy resources.

- Movement: If the sensor can be relocated between various objects or the object moves, it has been considered mobile. This movement will affect the efficiency, power source, and frequency range.
- Reporting frequency: This is dependent on how frequently the object can send the calculated data or collected parameters. So, faster data speeds like transmitting data per second would be more costly as they consume more resources.
- Report range allows calculation of the object's distance from the Gateway.
- Object density per cell of how many smart objects are in proximity to the Gateway with the same physical characteristics.
- Object data can be more straightforward, like the data transmitted by the humidity and temperature sensors, while richer data can consume more power.

3.5.2 Actuators

Actuators receive signals in digital formats such as an electrical signal or a binary signal and then cause an action-like movement (18). Figure 4 shows the process.

Figure 20: The interaction between sensor and actuator (17)

Sensors calculate and control the ambient environment and relay signals to the Processor for further processing. The CPU then sends commands to the actuator to take appropriate actions about that signals. Actuators come in several shapes and sizes (Figure 20). Actuators can be either high power or low power, depending on the type of motion they accomplish or the amount of power they can consume. As well as having two distinct stable outputs, they can also be continuous.

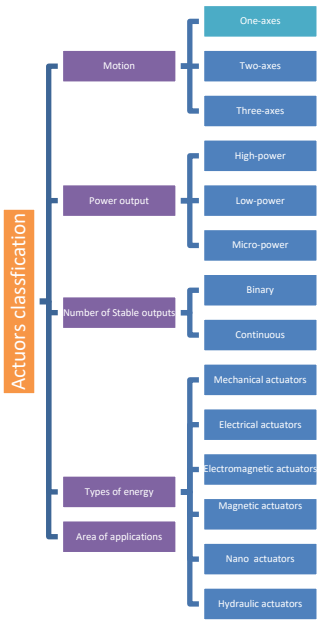

Figure 21:Actuator classification

3.5.3 Network Technology

We talk of smart objects collaborating with other smart or networked objects. The technology is focused on artifacts, needs, and use cases (18). A small mobile device would not need a powerful antenna and electricity—figure 6 displays the technology and use cases. The network will only be provided in small frequency bands (18). The IoT network is based on the range between smart objects and the collector. So, access groups can be as shown in Figure 22 and Figure 23.

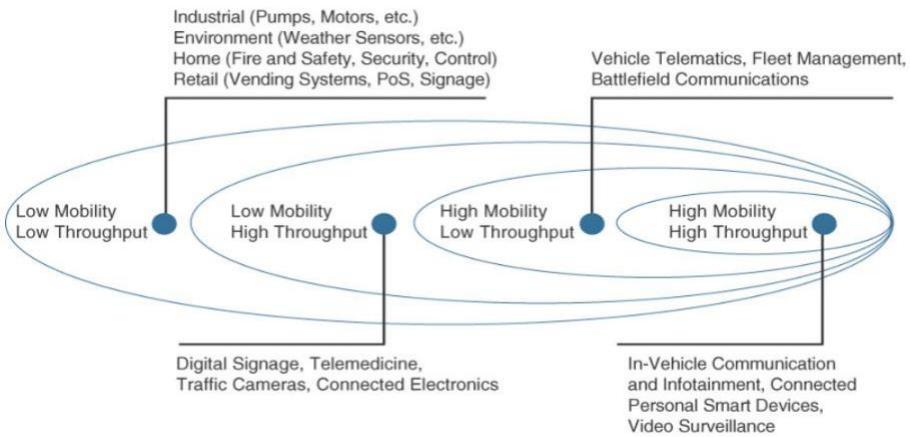

Figure 22: Network technologies (18)

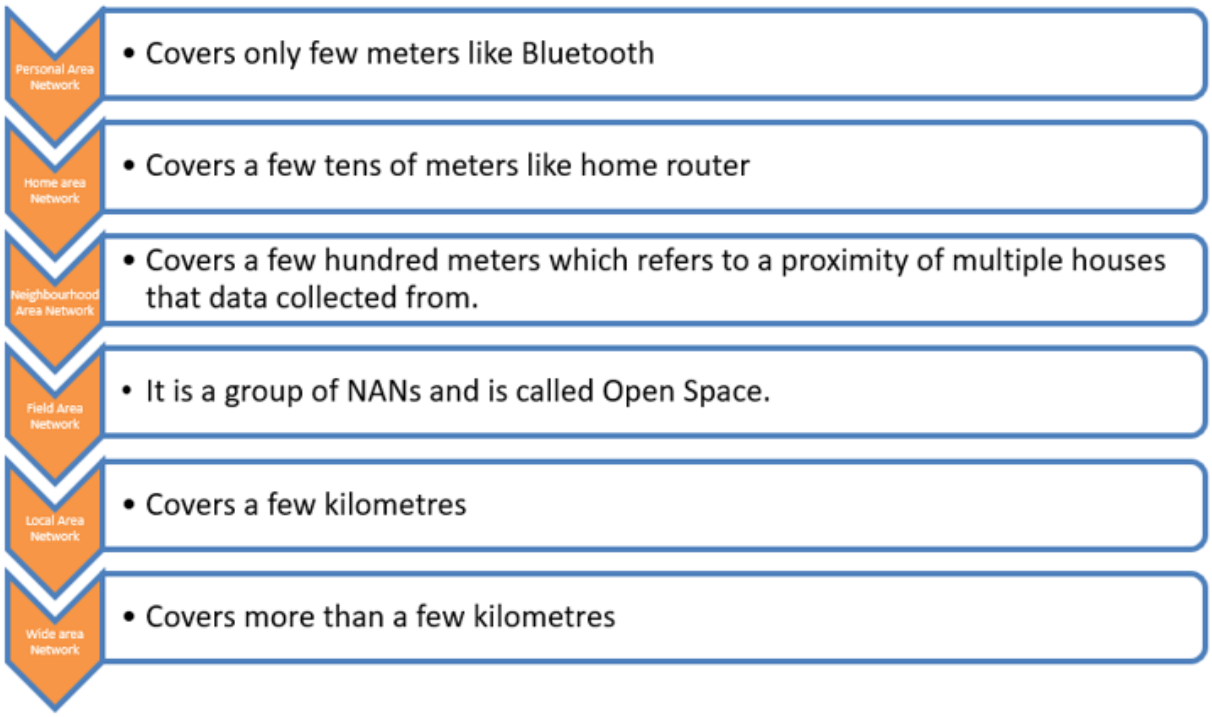

Figure 23: Network Classification

The number of smart objects of small proximity should be specified based on the transmission range, frequency, and data transfer rate. The number of wireless cells will be dependent on how much information needs to be transmitted and how much energy is required to make it move. That information. So, networking topologies for IoT can be:

I. **Point-to-point**: This topology is not a popular choice for IoT since only a single object can communicate with a single Gateway.

II. **Point-to-multipoint**: It permits the Information exchange between various equipment. Establishing a standard topology for different IoT devices helps communicate information between different objects. Some nodes will collect information and forward the data, while others will collect the information and forward the output. Further subcategorization is possible depending on the functionality of the node and sensor. In other words, some nodes can function only on the Core IoT Function Stack while the others operate on both the Core IoT Function and Data Management Function. By way of example, 802.15.4 standard (ZigBee) incorporates this characteristic.

III. **Mesh topology**: This topology will accommodate more nodes and allow them to interact with each other. The topology provides redundancy because multiple routes exist in each field since all nodes do not communicate directly. Full mesh is not standard in the IoT space.

3.5.4 Sensor Networks

SANETs, or Sensor/Actuator Networks, are networks of sensors to collect the data and information of their surroundings, then associated actions or functions are triggered as needed. An example of a SANET can be a smart home connected to sensors that can monitor heating and cooling equipment based on the received temperature.

In comparison, WSNs are networks for smart objects with sensors connected without a physical connection. Smart objects are wireless and linked to a self-organization network with complex topology (69). Sensing nodes consist of components transceiver(the combination of transmitter and receiver), sensing component, a processing module, and power. A sensing system consisted of various sensors and a digital to analog converter in which sensor readings will be converted to digital. A network port transmits data to other networks. The power module is either a battery or a solar cell Figure 24 illustrates the basic WSN structure. Soon, WSNs can forward sensed or monitored information to a computer using multi-hop routing. When the node needs to move, it uses a mobilizer component.

Another significant characteristic is that WSN can be disrupted rapidly .when communicating by air. Wireless Sensing Networks are low in cost, scalability, and reliability. However, WSNs face many difficulties and limitations, including low processing speed, latency, limited memory, low battery life and poor network connectivity.

Wireless Sensor Networks may include sensors of various types, which will be used to perform heterogeneous tasks. Communication protocols that can provide a trade-off between power consumption, resource usage, protection and throughput are required.

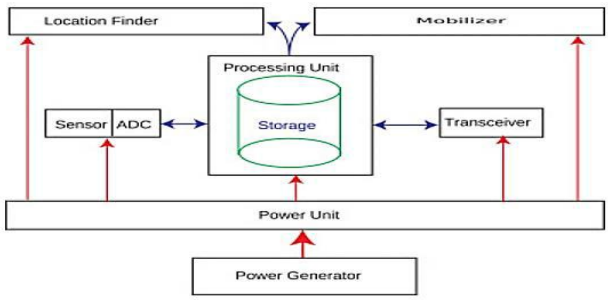

Figure 24: Basic components of WSN nodes (19)

WSN of individual sensor nodes gather information from different locations and then move the information to a central location for further distribution. The question is which topology will fit well with WSN from a topological perspective. All sections should be connected to the central node, and it should protect the whole area to make it fail-safe. For star topology, it is not practical as a single switch/hub failure will fail the whole network. Even this sort of topology is not appropriate for IoT applications.

One reason sensor nodes or smart objects can be inefficient is that they are restricted in resources. If a base station is far away from a set of nodes, then mesh topology could be implemented for such networks. In a topology where the nodes provide routing, the base stations have more bandwidth and consume more power. Next, hybrid network topologies were invented, which featured nodes with no resource limitations. So, these nodes can forward and relay data for multiple nodes, but only on their own. Instead, they must forward the signal to the unconstrained base stations or edge nodes (20), as shown in figure 25.

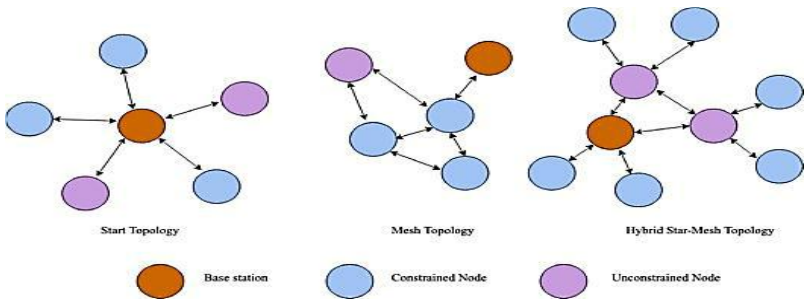

Figure 25: Comparison of different wireless topologies (20)

3.5.5 Sensors

Sensors gather information from the real world and send it to central locations such as Gateways, where the information is processed and used for further study. Sensors are generally inserted into an object connected to the cloud (18) (68). Sensors can be categorized according to their expected application range, as shown in Figure 26.

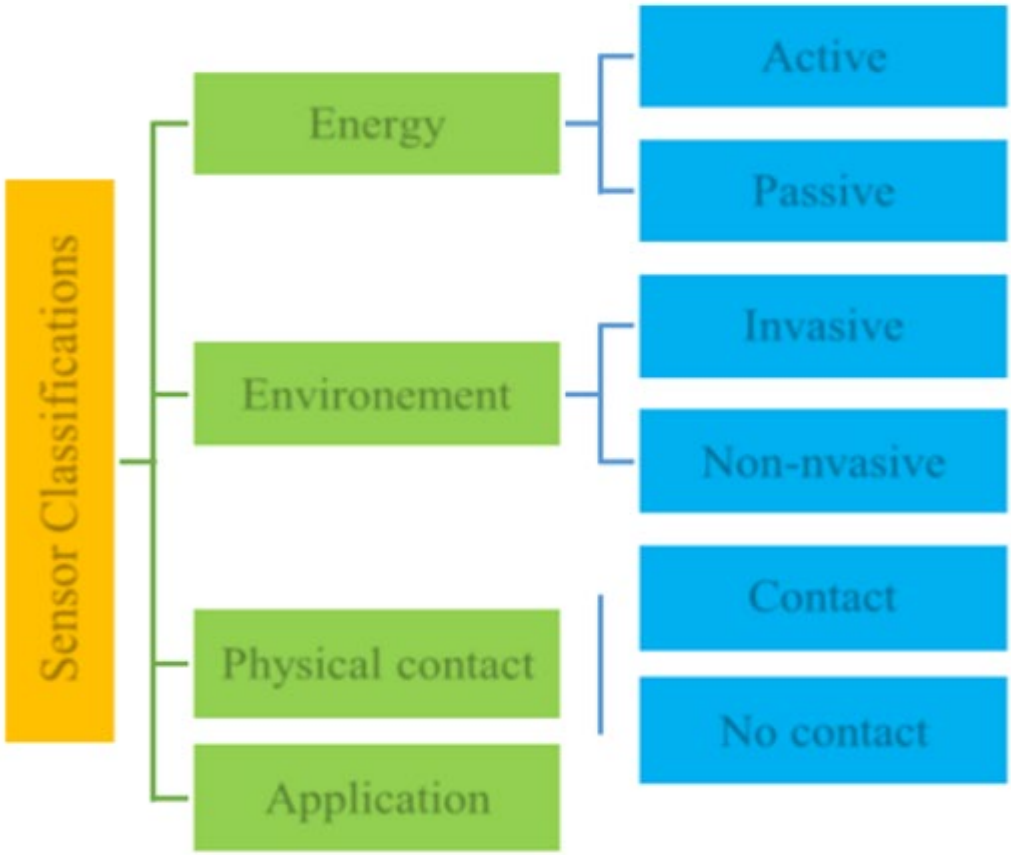

Figure 26: Sensors Classification

Active or passive sensors focused on whether the sensor generates energy or power as an output. If the sensor is active, it had an internal power supply or else, it is a passive sensor if it needs an external power supply. Invasive sensors are those that are inserted into an atmosphere and monitor a phenomenon within the environment. Some sensors need physical contact while measuring, and they are called contact sensors.

Finally, it can be divided based on the application type. However, there are many classifications. For example, some sensors are position-based, which measures an object's position; occupancy sensors that check humans, animals, or velocity sensors check and measure the object's speed or rotation per cycle, humidity temperature, and pressure sensors (18).

3.5.6 RFID

RFID is a technology that is embedded in the IoT for recognizing individual things. RFID "hard tags" replace bar codes and enhances the protection of goods. This technology enables the ability to recognize and object tracking. If radio tags are mounted in the units, they can be monitored using computers.

It consists of a reader and a tag, as shown in Figure 27. Tags have information about the objects and are associated with goods or things to allow items to be automatically recognized and monitored by radio waves. The reader will read the data in the tag and transfer it to the host for further processing.

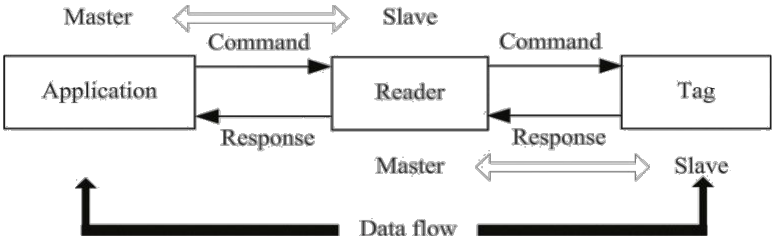

Figure 27:The components of RFID (21)

Here, RFID tags are attached to the objects, and they come in two modes: active or passive. The active tags are battery-driven and can express their state (on or off) without user interaction. Passive tags would be powered up by devices such as tag readers. Tags include wireless transmitters and magnetic strips that store data.

The reader has a radio frequency interface and software to relay data to the tag's receiver. The application program is a database or an application that can do the data processing when the user scans the tag and initiates the contact between the tags and the readers. However, the need to use RFID tags brings a few security problems, and there will be a risk to eavesdropping, spoofing, traffic analysis and DDOS attacks. Besides, privacy may be adversely impacted due to unauthorized read of the access control plans. For instance, getting location information by predicting the tag responses (21).

3.6 IoT Testing

Quality assurance testing in IoT is a way to discover whether the Internet of things Technology is of decent quality and meets user expectations. Testers must rigorously test the product by looking for and correcting any underlying bugs to ensure product consistency. There is a wide range of tiny, interconnected devices that harmoniously work together on a network. Because of the minor bugs in different systems, it is impossible to identify the failing system's exact cause. The efficiency of each system influences the output of all devices. IoT testing ensures that all its features are implemented effectively by each methodology, ensuring that the entire IoT network operates together and works as expected. The following are some areas where IoT innovation can positively impact applications (70).

3.6.1 TYPES OF IOT TESTING:

Data derived from various sources, including sensors, inventory, machines, tools, people, and production logs, can be inaccurate due to multiple errors. Here are the different testing procedures that are essential when testing the IoT (70).

Performance Testing:

The customers have become more susceptible to minor glitches. Furthermore, when they face even the slightest delays, these clients of IoT devices stop using the applications. That is why performance is imperative to evaluate the performance metrics of the IoT application. Performance measurements for these IoT devices include database read/write speeds, loading time, throughput, performance, uptime, and data transfer rates. In a networked application, the end-to-end approach becomes even more critical. Performance testing can be used to determine how well an IoT device app performs during regular use over long periods and how it copes with the increased load during peak times.

Security Testing:

Although security with IoT devices is perhaps one of the most significant user concerns, it is absolute. The Internet of Things has blossomed into enabling various mobile, smartphone, tablets, laptops, and desktop applications that allow various users to dial in real-time, digitally share surveillance and health information online.

It is allowed for most apps to have the ability to read or access private data inside an IoT device, so these apps should have some specialized access controls for users that restrict access for external people. Without performing security testing, it is impossible to know if there are security

vulnerabilities in an IoT software code to protect against leaking sensitive information to an attacker.

Database Testing:

Since they are stored, databases allow IoT devices to operate and are essential to their working, which is the most significant advantage of testing wireless connected IoT computer applications. The technique includes checking for various kinds of errors that exist during query execution. Data to be collected must be complete to be processed, and there should be no changes to the stored data during its processing. We need to monitor if the application being tested is responsive. Data shown in its interface is the actual data displayed in their app and ensures that all query's response time is optimal.

Functional Testing:

Every computer being made for IoT systems must execute several different functions. In defense of these IoT system applications, it is essential to evaluate each of them separately such that the core logic(high priority scenarios) is tested after every deployment. It is mandatory to operate the IoT systems 24*7, i.e. environment with zero downtime, so a detailed regression test(test activity) is needed to achieve the above conditions.

Compatibility Testing:

A compatibility assessment is intended to check an IoT computer’s compatibility with various device formats, network software, and operating systems. Compatibility testing is responsible for determining if the IoT software app is responsive and running on different devices and browsers to provide the optimal user experience.

For example, when creating a smartphone application for an IoT device, it is a reasonable thing for developers to make sure that their interface features a way to print the image. The app should be compatible with all printer types to prove their need to test the app compatible with all major printers.

Usability Testing:

To test an IoT interface app's usability, we can perform a user-testing session, which measures whether a functional system provides the desired user experience to a typical consumer. Therefore, usability testing is a continual procedure that touches upon many facets of a user interface, including the ease on which the user may navigate various devices, the usability of the device's features.

Scalability Testing:

IoT devices are gaining attention daily, and with the growing social isolation from other citizens, we should anticipate IoT devices to become bigger and more frequent. To measure the strength of an IoT app, we should do Scalability testing. That means we can see how to deal with and can accommodate many users without slowing down the system.

Reliability Testing:

IoT devices are gaining attention daily, and with the growing social isolation from other citizens, we should anticipate IoT devices to become bigger and more frequent. A Reliability testing guarantees whether an IoT app will handle up to any number of concurrent users while retaining the same quality of efficiency.

Network Testing:

Without a smooth network link, IoT apps cannot function efficiently. IoT software can be checked on its capacity to manage multiple network links and connection protocols to be utilized across an IoT framework.

Pilot Testing:

Real-time testing of IoT devices is not sufficient since an application based on these devices can fail when exposed to real-time conditions or data. The program was placed into action in a small range of setups, where each session member's behavior is monitored. These participants use the program and provide feedback about how it can be enhanced and adapted. The testing process is a crucial step for developing a stable and suitable product for production implementation.

Regulatory Testing:

One way to conceive this technology is to imagine medical instruments are wireless. Nevertheless, this business is prone to several regulatory restrictions. Building systems that communicate with healthcare data in innovative ways enable teams to navigate through many enforcement tests.

Even if an IoT system is ideal for practice, performance, protection, compatibility, and usability metrics, it will fail if it does not meet HIPAA enforcement criteria (Health care Apps). It is best to familiarize developers with legislative criteria, aspects, and regulations at the planning period's outset.

Upgrade Testing:

IoT systems are made up of many communications and networking techniques, utilizing various protocols, applications, hardware, firmware, and modules. Whenever new parts and prototypes are introduced, they need to be retested to ensure they all function as expected.

3.6.2 IoT Testing Approaches:

IoT devices, ingenious devices, have a different design than most other devices. When IoT worlds become more and more integrated, the opportunity to assess this more comprehensive device scope is critical. In an IoT platform, data collected by the sensors are passed into the back-end servers. Afterward, it is sent out to distant nodes for processing. Each part of the Internet of Things has various testing procedures. When they are running these experiments, testers should consider formulating a comprehensive test strategy.

IoT Security:

To find any unauthorized access in the systems and track systems status, one must ensure that they have functioning security protocols. Defending against network attacks should be the priority, and it should be looked at as a high-level issue. Often, there is a range of testing done to ensure the protocols are met. Based on the system, security protocols should be designed and implemented in the lifecycle or system maintenance. One needs to build a secure testing environment to see if the data is secured while moving from one computer to another.

Usability:

This test is used to ensure ease of using the system under all possible conditions; nowadays, corporations are working on their ecosystems and integrating them into that. Stable IoT systems should able to work under all types those ecosystems and able to use the application with or without basic user training,

Pilot testing:

In-place calibration of IoT is often one of the procedures to be performed before sensor deployment. Testing a particular device or a framework is testing the component or system's usability based on outcomes in a real-world scenario to examine its output and debugging bugs or issues. The word "pilot testing" is often an equal term.

Cross-domain compatibility testing:

This program allows keeping track of all the computers and their Internet access. Ensuring that the correct protocol for research is followed from the technologies like Windows XP to PGP encryption, security problems, such as public/private critical collections, permits, and access control lists are maintained during the research process.

3.6.3 IoT software testing tools

Different tools allow for the testing of both software and hardware components independently in IoT. Most of the tech testing methods using are (71):

Wireshark:

It is a free network trace application that is open source & free to use. It is used above what can be seen to track all the traffic visible on the interface, even unicast traffic. In Tcpdump, there is a command for displaying packet traces with a graphical user interface and support for file sorting and filtering. It is Another edition without an Interface is recognized as TShark. Like bash, it operates on various operating systems, including Debian, BSD, Solaris, Microsoft Windows, and macOS.

Shodan:

Shodan is an IoT monitoring application that can discover which computers are already wired to the Internet. This acts as an assistance method to maintain track of all the devices that are related through the Internet. Each machine is linked to a network of computers upon the Internet.

Tcpdump:

Tcpdump performs similar jobs to the ones wire shark tool does, but it does not have a GUI. It is a command-line application that shows all TCP/IP packets and other data over the network as they are sent and received.

Thingful:

Thingful is a website for all products that go online. The Internet of Things (IoT) promotes the safe interoperability of billions of items over the Internet. This IoT research platform also helps the consumer monitor how data is used and empowers them to make more decisive and valuable decisions.

SOASTA CloudTest:

Under its features, the method supports four distinct forms of automation: Handheld functional and performance monitoring, Web-based functional and performance testing, load

testing, and testing for capacity constraints. Millions of globally distributed simultaneous users are checking the program in browsers under tons, close to Facebook. With real-time analytics, smooth integration between Test Design, Monitoring & Reporting, & real-time Online Reporting, CloudTest allows easy and rapid testing.

3.6.4 Technology/skills vital for effective IoT Testing

There is always a shift in technology trends to test complex IoT applications tester that needs to be updated with the latest techniques for efficient coverage of requirements. Some of those skills are (72):

Machine Learning:

If we gather vast volumes of data, it will make sense to examine the trend and check for possible outcomes. When more things become linked and knowledgeable, AI would be called upon to automate several functions. Will be explored briefly on this topic in upcoming chapters.

API Automation & Testing:

An MIT study reveals that 40% of IoT's importance is in the willingness to use it. Development teams that understand how and when to link automated API testing with repetitive tasks developed comprehensive data, companion testing methods, and interactive frameworks will bring their products to the market.

IP Networking:

In wireless sensor networks, embedded sensors communicate with their surroundings. The details are shipped off for further review. The system that links knowledge must be flawless, ready for large traffic volumes, and safe and stable. Developers should know the principles of OSI, how networking protocols operate, and the existing IoT communication requirements.

Business Intelligence:

It is essential to revisit the more expansive IoT issue space to grasp its present reach. It is all connected to collecting, saving, and analyzing data obtained from smart devices. If not grasped the significance or meaning of results, how will we perform research? Previous experience with processing sensor results, operating a data center, predictive analytics, and programming in Hadoop and NoSQL was deemed positive.

3.7 Benefits of Internet of Things

In our career and personal life, IoT persists in having an impact. A few of the IoT advantages are subtle, but it is impossible to miss others. Six ways in which individual will benefit directly from the Internet of Things are:

1. Connectivity: Bid farewell to the period where different devices, i.e., manually managed. Welcome to the ability to operate multiple things from one device, for instance, a smartphone. Each device will soon be connected to one other for streamlined control, from controlling the thermostat to turning up the TV volume to dimming the lights and more.

2. **Efficiency**: Increased connectivity means reducing the amount of time spent carrying out the same tasks. For instance, without needing to pick up the phone or turn on the computer, voice assistants such as Apple's Homepod or Amazon's Alexa can answer questions. As they can quickly provide essential updates and information, they can even eliminate the need for many business meetings.

3. Convenience: Smart devices, particularly at home, are becoming more commonplace. Smart refrigerators and Amazon Alexa are a few examples of IoT devices that make reordering items simpler, requiring little more than an action or two signaling consent. These advantages of IoT can save time and make life simpler.

4. Wellness: Smart cities are also growing, and IoT developers are developing ways to monitor urban conditions using the IoT, such as traffic, air quality, electricity/water use, and environmental factors.

5. Conservation: Smart cities are growing, and IoT designers are developing ways to use the IoT to track city conditions, such as traffic, air quality, power/water use, and environmental factors. Doing so can help to find solutions to current problems and conserve resources for city planners and residents.

6. Personalization: More individualized interactions are better relations," as customized connections are "more appropriate, more intriguing, less irritating, and more pleasant. As IoT devices collect more information from the individual, they learn likes and dislikes quickly and tailor their services to preferences.

Chapter4

IoT Connectivity protocols

IoT devices are typically resource-constrained, and information loss or a high memory requirement may involve stringent requirements to provide effective communications. A vital component of the IoT technology stack is the IoT protocols. Without them, the technology would indeed be deemed pointless because the IoT protocols enable it to exchange information in a structured and meaningful way (73). The vital feature that makes the Internet of Things is the interaction between access points, sensors, gateways, servers, and application programs. The IoT protocols enable all these smarts objects to be spoken and conversed securely, which can be seen as the IoT gear's syntax to interact; out of transferred pieces of data, information can be extracted end-user. Thanks to this, the entire deployment, especially IoT device administration, is economically profitable.

4.1 IoT Information Communication Technology

ICT is considered an evolutionary leap transmission of information. If connected, these "smart devices" will communicate with each other, receive and react to the news, make decisions and take action in the physical world. The new technology being applied is called connectivity for everything. The IoT environment consists of the world's most popular smart mobile devices and wearables but many limitations. Unfortunately, the storage capability, power life, and radio range are all considered shortcomings of today's digital radios. Therefore, it is essential to efficiently enforce a detailed IoT communication protocol to efficiently handle these conditions (74).
In IoT for connectivity, several factors and differences need to be considered (18):

- **Range**: IoT is based on the short-range wireless method and medium-range wireless method. Simultaneously, short-range communications (SRD) function like serial ports, Bluetooth and IEEE 802.15.1; they do not cover much distance. Next-generation interconnectivity belonging to IoT (Internet of Things) technologies. It can cover farther distances. The example may be IEEE 802.11 (Wi-Fi), IEEE 802.15.4, and IEEE 802.3 (Ethernet).
- **Frequency band**: The IoT spectrum's frequency bands can be separated into licensed and unlicensed bands. When using licensed bands, the issue becomes more complicated since we cannot use them unless we pay to have access and the standard of service is mediocre. The ITU then launched its unlicensed ISM bands for IoT connectivity, including 2.4 GHz, tailored to Wi-Fi technology. Bluetooth IEEE 802.15.4 Wireless PAN is intended for low energy IoT connectivity; it outstrips licensing costs and allows it to be used without a service provider.

- **Power Consumption**: In IoT, the nodes are divided into a powered node and a battery-powered node. Powered nodes (the kind that carries USB cables) have the power source right next to them. Businesses that rely on these devices should not restrict their energy supply unless they can use less energy, although they struggle with flexibility and mobility. On the other hand, battery-powered nodes depend on their internal battery's lifetime. nodes need to use low power consumption and innovations like Low Power large Area Network (LPWA) technologies.
- **Topology**: The networking in some technologies (such as IoT) works based on a star, mesh or peer-to-peer topology. Star topology is better for a long-range two-way wireless mesh network, cellular networks, or a Bluetooth network. A controller will be provided to provide connectivity among the endpoints. As for long-range wireless technologies like IEEE 802.15.4, they can be implemented in mesh and peer-to-peer topologies.
- **Constrained node networks**: It refers to low-power and lossy networks. The device's radio interference can affect the network speed and cause a dip in the signal. Where energy is a limit. Therefore, Layer 1 and 2 protocols should be able to satisfy these restrictions.
- **Payload overhead**: Since IPv6 is all about moving information around, the minimum payload MTU size for IPv6 is 1280 bytes. In networks with smaller Maximum Transmission Unit (MTU), fragmentation Link Layer in the Connection Layer are created.

4.2 Connecting IoT Objects

There are many things about devices out there that need to interact with each other. Since the networks are wireless, the networks allow for networks of networks to be built and implemented. There are many wireless innovations on the market, but the most popular wireless standard is the IoT platform-optimized 802.15.4 standard. The object of this section is to express an overview regarding the 802.15.4 family of protocols. Sensor networks are connected, and a controller connected to a cell phone network contributes to other criteria (18):

- Power limitation: Sensor power limitations exist. In order to be independent and live a long time, a battery must excel at battery efficiency.
- Topology changes: The network topology may change by adding new devices or changes in the connection type. So, the topology needs to be flexible. Communications would be low-intensity and low-powered. Therefore, the throughput will be reduced.
- The data transmitted through the Gateway would be different depending on the network's size, the number of devices and the frequency at which they transmit data.

These specifications and the OSI model suggest that conventional protocols would not function efficiently in the IoT platform. The IoT platform requires a protocol that is low overhead and effective in energy usage. The sensor should be power-efficient because it needs to work on small power supplies. It should have stability and reliability. Since network topology and device placement can change over time, the MAC layer protocol should be scalable; hence, IEEE 802.15.4 protocol was invented.

4.3 IEEE 802.15.4 Standard

IEEE 802.15.4 asks about how a lower-layer protocol stack of WPANs can be used to achieve the aim of providing universal access at a low cost. Compared to older, more conventional approaches such as Wi-Fi, more bandwidth and higher power are needed. The emphasis is on low-power wireless communication of nearby users, which can lower total power consumption even further.

This technology provides easy installation and is very flexible and straightforward. It was then improved, and its range increased and could support the building's scale but not more extensive than that. Examples of 802.15.4 style solutions include home automation, automotive networks, and industrial wireless sensor networks (18). Figure 28 shows the IEEE 802.15 protocol stack:

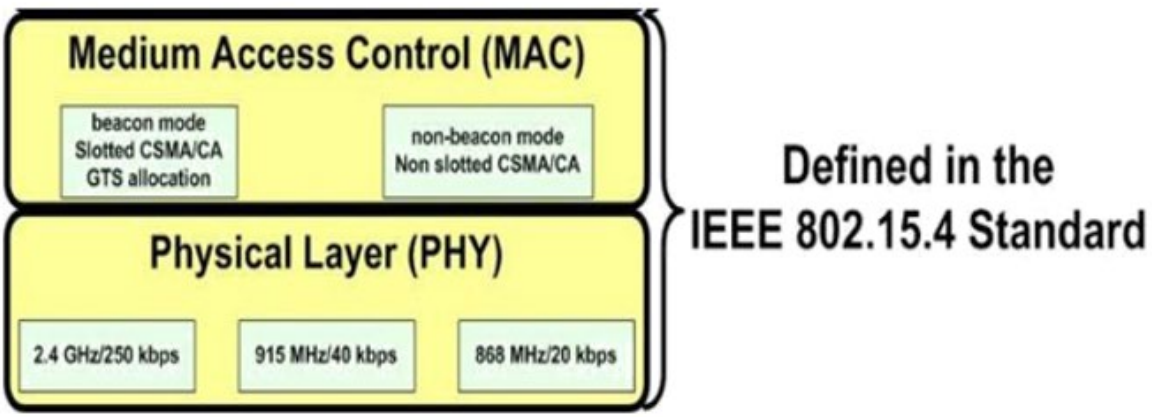

Figure 28: IEEE 802.15.4 PHY and MAC layers (22)

IEEE 802.15.4 was mainly designed for PAN and low-range networks, which brings out lots of IoT-friendly features (74).

I. Low data rate communication: IoT devices can not send heavy-load traffic, which requires a reasonably high data rate, and it is unlikely that a system sends traffic of 4 Gbps, which is more common in IEEE 802.11 standard.
II. Scalable: Implementing any "smart gadgets" will theoretically be done without a single intervention.
III. Low operational cost and power: In IEEE 802.15.4-based network system, a small battery device work somewhat. There are dozens of functionalities in the IEEE 802.11 standards, contributing to some protocol stack energy consumption. When a Wi-Fi system connects to

an access point, it creates a session. When linked to wireless Internet, it must sign in for an update. So, numerous frames exist in IEEE 802.11, which display the operations of an 802.11 network. This habit contributes to electrical consumption. If IEEE 802.15.4 standard is used, only low-power messages are sent.

IV. The network infrastructure is ideal for multi-protocol monitoring and control applications. It supports both mesh and star topologies.

V. IEEE 802.15.4 belongs to a group of networks that can be up and running at a single glance.

VI. Low Power Area wireless area networks are less reliable than Ethernet or fiber connections.

VII. The packet size is relatively small, and the MTU size in IEEE 802.15.4 is 127 bytes.

Beacon-enabled or non-beacon-enabled are access methods in IEEE 802.15.4 standard. BE Modes are used to manage power at various degrees. At each active timeslot, nodes will communicate with the coordinator. At each inactive timeslot, nodes will switch into the low power mode for conserving resources. At the same time, during contention cycles, TDMA will be used, and a timeslot will be allocated to each node. When the non-beacon mode is in use, CSMA/CA is used, and nodes remain involved, and there will be no conservation.

As for wireless sensor networks, devices in IEEE 802.15.4 are divided into two groups (75): Full-Function Devices (FFDs): Devices or nodes belong to this group are capable of performing all of the IEEE 802.15.4 functions and can also switch roles. FFDs can only speak among themselves. They are known as network coordinators.

I. Reduced-Function devices (RFDs): They have minute features RFDs can only communicate with FFDs since they have less computing capacity and less memory space than FFDs. They may not have a considerable number of data and can be associated with fractioned data. They are designed to perform simple tasks and are not intended for more demanding acts.

II. Full-Function devices(FFDs): A computer or node falls under this group, and it can perform all the functions of a compatible IEEE 802.15.4 transceiver. FFDs can interact with other FFDs directly. They are called network coordinators. It is also critical that devices in this category can be controlled in a variety of different modes.

   i. PAN coordinator: It is a PAN controller, and it can either build its network.

ii. Local coordinator: It connects to neighboring beacons to provide synchronization services. PAN coordinator can be affiliated with a local coordinator. The local coordinator cannot build networks of their own.

iii. End Device: The end device should be first associated with a PAN coordinator, then associated with a local coordinator.

4.3.1 IEEE 802-15-4 PHY Layer

IEEE 802.15.4 PHY layer is responsible for wireless data transmission in the following frequency bands and dynamic channel selection. Channel selection is based on Direct Sequence Spread Spectrum (DSSS) technique, and channels can be used for signal power, connection efficiency, switching channels, search for the beacon (75):

- 2.4 GHz, 915 and 868 MHz
- Ten channels between 902- 928 MHz (40 Kbit/s)
- The single-channel between 868-868.6 MHz (20 Kbit/s)
- 16 channels between 2.4.- 2.4835 GHz (250 Kbit/ sec).

Lower frequency bands are more practical to penetrate objects and walls; the 868-915 MHz range would better suit this purpose. Despite the maximum data transmission rate in the 2.4 GHz ISM band, IEEE 802.11b also operates in this frequency range, leading to interference and other problems (18).

IEEE 802.15.4 PHY layer responsibilities include:

I. Radio transceiver activation and deactivation

i. Radio transceiver activation and deactivation will happen based on the MAC sublayer request.

ii. Radio transceivers can function in one of these states transmit, sleep, and receive.

II. Energy detection

i. The receiver of the signal power will be tested.

ii. No signal identification and decoding will be made.

iii. Energy detection is done for purposes like selecting an appropriate channel algorithm or checking if the channel is busy or idle.

III. **Link quality detection**
    i. Assesses the quality of the received packet, which in turn indicates the signal quality.
    ii. Techniques like SNR (signal-to-noise) can be used.

IV. **Channel Assessment:** It checks whether the medium is idle or busy and has three modes.

V. **Energy detection mode:** if the energy is above the threshold, the medium is busy.

VI. **Carrier sense mode**: If a signal with modulation and characteristics of IEEE 802.15.4 is detected, the channel is busy.

VII. **Carrier sense with energy detection:** It is a combination of both approaches. Selecting channel frequency There are 27 channels for IEEE 802.15.4. The transceiver should be placed in one of the channels.

4.3.2 IEEE 802-15-4 MAC Sublayer

IEEE 80215.4 supports two media access control methods:

Beacon-enabled: When the PAN coordinator chooses this access process, a beacon frame will be transmitted regularly by the PAN coordinator or local coordinator for synchronization. Furthermore, medium access is based on the contention-free timeslots and the slotted CSMA/CA. Contention-free timeslots will be reserved and considered for specific time-sensitive nodes and applications requiring QoS and guaranteed bandwidth. Beacon frame will be shared between nodes and over the medium, which means that the frames are not very wide.

Nonbeacon-enabled: In this mode, no beacon frames exist, and CSMA/CA is employed for the medium access methods in IEEE 802.11 Wi-Fi. Initially, IEEE 802.15.4 did not support network-based applications. In this version, the PHY Layer was changed to allow for wider PHY operating ranges.

4.3.3 IEEE 802.15.4 vs IEEE 802.11 and IEEE 802.11ah for IoT

Wi-Fi systems are one of the most appealing of our IoT technologies. However, Wi-Fi is not capable of accommodating unlicensed sub-GHz signals. sub-GHz nodes run in a very low-power band of millimeter waves. The IEEE 802.11ah standard was established to mitigate this problem. This standard allows some applications and promotes the implementation of sensors and meters in smart grids. It also allows expanding Wi-Fi communications beyond the scope of the current IEEE 802.11n standards, which overlap with IEEE 802.11n standards. It also supports aggregation of the IEEE 802.15.4g subnetworks and several other uses. The Wi-Fi Alliance ratified a new feature to support very low frequencies (18).

Besides, the IEEE 802.11ah standard appropriately incorporates the best aspects of both low-power sensor networks and Wi-Fi. This specification includes single-hop contact over at least 1000 meters, and relay access points are used to expand the connection to nodes that are multi-hop away. 802.15.4 alone cannot allow for extensive contact distances.

The original IEEE 802.15.4 standard operates within the unlicensed ISM band for up to 250 kbps, while the new IEEE 802.11ah standard uses the Sub-1GHz band for having a better range. The traditional IEEE 802.15.4 standard failed to deliver collision avoidance, per their contention-avoidance process. On the other hand, IEEE 802.11-based WLAN operated effectively using a RAW-based technique (76). It allows for scalability by allowing large numbers of nodes to be linked with one AP.

	IEEE 802.11	**IEEE 802.11ah**	**IEEE 802.15.4**
Frequency bands	2.4 GHz- 5 GHz	Sub-GHz	2.4 GHz and Sub-GHz
Channel access	CSMA/CA	RAW	CSMA/CA
Range	Medium-High	Medium	Medium
Topology	Adhoc-infrastructure	Star	Star, Mesh
Data rate	Medium-High	Low-High	Low

Table 4: Principal features of IEEE 802.11, IEEE 802.11ah and IEEE 802.15.4

Generally, Wi-Fi can be used in different IoT testbeds since it is compatible with IP networks and is incorporated with existing infrastructure. IEEE 802.15.4 is designed for small, ad hoc wireless networks with limited power and data rates. Network analysis using IEEE 802.11 does not perform well and is inefficient. The study reveals that it is advantageous of IEEE 802.11 regarding packet transmission, throughput, and end-to-end delay. IEEE 802.11 can achieve a better packet delivery ratio than IEEE 802.15.4. Increasing the number of nodes can make a radio network more susceptible to interference (77).

4.4 ZigBee

ZigBee is one of the wireless networking protocols that incorporate IEEE 802.15.4 standards, bringing additional functionality to IEEE 802.15.4. It is a low-cost, low capacity, low bandwidth wireless protocol for low-powered battery-powered devices. ZigBee's lower data rate is well suited for low data rate situations like wireless communications between sensors and controllers (18).

ZigBee Alliance Community developed this protocol in 2003; it uses the IEEE 802.15.4 MAC and PHY layers, introduces additional features to obviate the shortcomings and offers logical network, security, and application applications. The wireless mesh network is based on self-organizing requirements (78).

Wireless devices compliant with the ZigBee standard can work with one of these frequencies: 868 MHz, 915 MHz, or 2.4 GHz[16]. It has valuable security features. The network and security layers create the network, configure routing tables, calculate routing routes, evaluate neighbor discovery, set up topology, and provide security.

Concerning confidentiality, it utilizes the IEEE 802.15.4 security solution implemented in the MAC layer and utilizes AES 128-bit encryption. It also applies protection to the Application and Network layers to avoid intrusion (18).ZigBee is a wireless protocol platform that is used for peer-to-peer networks and tree networks. In particular, the ZigBee protocol is based on the topology of the star, the tree, and pair to pair.

There is an ongoing partnership between ZigBee and the associated IEEE association to ensure that any required development standard is met. ZigBee is available in all seven OSI layers. In ZigBee, many industry-specific applications exist, and some vendors tailor it to their particular use. This industry includes smart energy, security systems, medical data collection, utilities monitoring and control, and home automation.

One example of the ZigBee application can be used in home-based monitoring systems to track blood pressure. For the remainder of the analysis, the information will be collected through a ZigBee-enabled wearable system. The patient's data will be sent to a central device like a patient's electronic computer. The patient's vital information will be sent to their nurse wirelessly and via the Internet (18).ZigBee only offers interoperability for devices that abide by the ZigBee standard (18) (79).

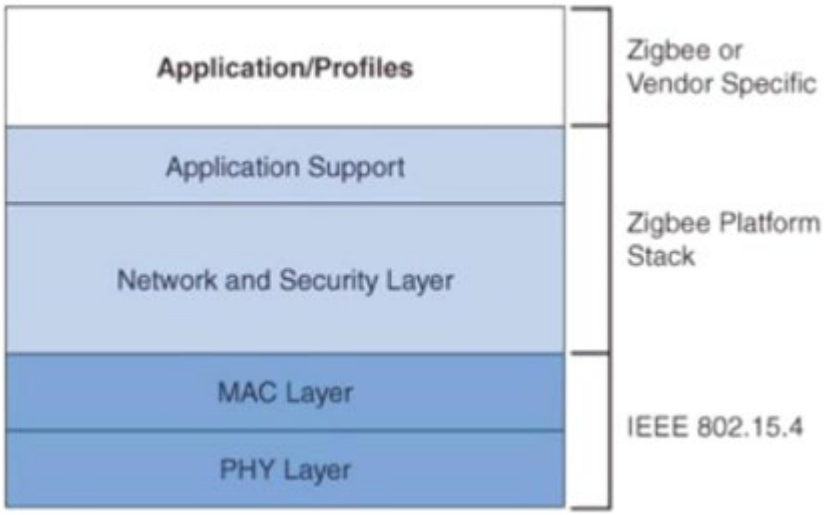

Figure 29: Zee Protocol Stack (18)

The IEEE 802.15.4 is becoming the ZigBee protocol's foundation; it focuses on two layers of the OSI model: network and application layers. The following is the characteristics of the ZigBee protocol (79) (78):

- Implementation of ZigBee and IEEE 802.15.4 in a wireless network is more straightforward and more petite than IEEE 802.11.
- Low energy usage makes the battery life of at least two years for devices.
- It is cheap because the protocols have been reduced, and the data rate has been reduced twofold.

- Low offered the message throughput and has a time, sporadic transmission.
- Lower data rates and are comfortable in their service.
- Support wider network.
- The transmission scheme is DSSS.
- No reliable quality of service.
- Networks appear to be protected through AES 128-bit encryption.
- It offers a reactive routing system to balance supply and demand by constructing routes instantly.

Also, Zigbee products use IEEE 802.15.4 along with DSSS1, and QPSK ((Offset-Quadrature Phase-Shift Keying)offers an outstanding output (78) (80) (81)

- The mesh networks link via several hops, creating an ad hoc network, hooking up to public networks for automated route finding, and healing its faults.
- By listening to the data on the network before sending, CSMA/CA access method is employed.
- The acknowledgment per-hop means that the retrial can be performed up to a maximum of three times on each failed packet on a communication. If the packet contains a failure, it will be sent out to the sender, and the node will notify the sender.
- 16-bit CRC frame checksum.

ZigBee's limitations are (given) the fact it is a single connection technology. It lacks in the issues above, it makes up for in being closely coupled with application profiles, shortcomings in integrating with the Internet, and has one of the highest data rates in the industry.

Physical Layer

In ZigBee, the PHY layer is IEEE 802.15.4, the radio transceiver responsible for communicating and controlling the radio transceiver. Tasks include radio monitoring of packet routing, channel selection, modulation selection, and transmission speed requirements (74).

MAC Layer.

MAC Layer is responsible for the association of services and disassociation of services, and if the system is the coordinator, it creates a beacon frame. Beacon delivery provides prompt schedules for all nodes (74). However, using a beacon adds extra difficulty to a ZigBee deployment. MAC layer manages channel access to prevent collisions. When a node wants to transmit, it tests to see if the transmission medium is open, and if not, it waits until the medium is free once again. This time is specified by the timer and backoff algorithm, which retries transmission until receiving data successfully. ZigBee protocol rewards for all ZigBee packets in a hop-by-hop format. There are four MAC frames, namely:

- The beacon frame is used to synchronize the clocks on all the network computers. The computer is informed whenever data is pending for it. If the system does send data to a coordinator, it will do so through indirect means. It is used to transmit data.
- Data frame is the network layer's payload. Next is the MAC layer payload.
- Acknowledge frame accept the contents of the data frame. It does not have a malicious payload.
- MAC frame: Commands like the request to associate/disassociate can be referred to as MAC frames.

Network Layer

Much like an IP network, it performs routing and addressing. At each destination, ZigBee coordinators and ZigBee routers can perform route discovery. The coordinator and address assignment will specify the network type, and the coordinator will do topology.

ZigBee provides two primary types of routing: mesh routing and tree routing., mesh routing requires running a data line in a point-to-point fashion between the ZigBee nodes. Additionally, Ad hoc On-demand Distance Vector (AODV) in routing responds to a destination in an on-demand manner. When routes are required, they are created. When the coordinator is the root resource, all devices are the leaves. However, this routing is ideal for small to medium-sized networks, but resource constraints cause memory overflow in complex networks (74).

Application Layer

User data resides here. Program artifacts monitor the Ethernet and USB interfaces in ZigBee devices. For a particular application, an application profile can be identified, and the processing actions can be specified.

There are two distinct forms for a user's device profiles, either vendor-specific or public-specific. A public profile is designed to work with any vendor's products, while a vendor-specific profile is designed to work with only one vendor's products (74).

4.5 CoAP Protocol

The Internet of things runs over the weak links(low power) between devices. A TCP-less Internet, such as the ones we have today and our ancestors had back then, is reluctant to use a connection-oriented protocol like TCP, such as the Internet's transport layer. Internet Protocols, such as HTTPS and HTTP, are a session-oriented protocol and use a TCP. Another protocol called CoAP(Constrained Application Protocol) is designed for IoT applications to overcome legacy protocol's limitations (82).

The CoAP is an IETF (Internet Engineering Task Force) protocol from IoT that serves as HTTP's counterpart in the IoT. It is a lightweight web transfer protocol that is designed for use on a limited resource device. The protocol that comes in over UDP and uses a low overhead. These are devices or, in other words, IoT devices that are designed for stationary services. The Representational State Transfer architecture(REST) is used to manage this protocol. This protocol facilitates the communication between smart objects on the Web.

CoAP is a reasonable way to present the data needed by RESTful application as HTTP packets. CoAP packets are smaller in size than HTTPS/TCP packets. A subset of CoAP requests includes GET, POST, PUT and DELETE (24) Protocols like HTTP and HTTPS; the client communicates with web entities through web browsers by exchanging requests and responding to HTTP messages (24)

CoAP also adds a few extra features, like the ability to be lightweight, allow a few features to be sent back and forth without a lot of excess data, provide a framework for discovery of resources, an asynchronous messaging system (24) Figure 30 demonstrates the CoAP architecture.

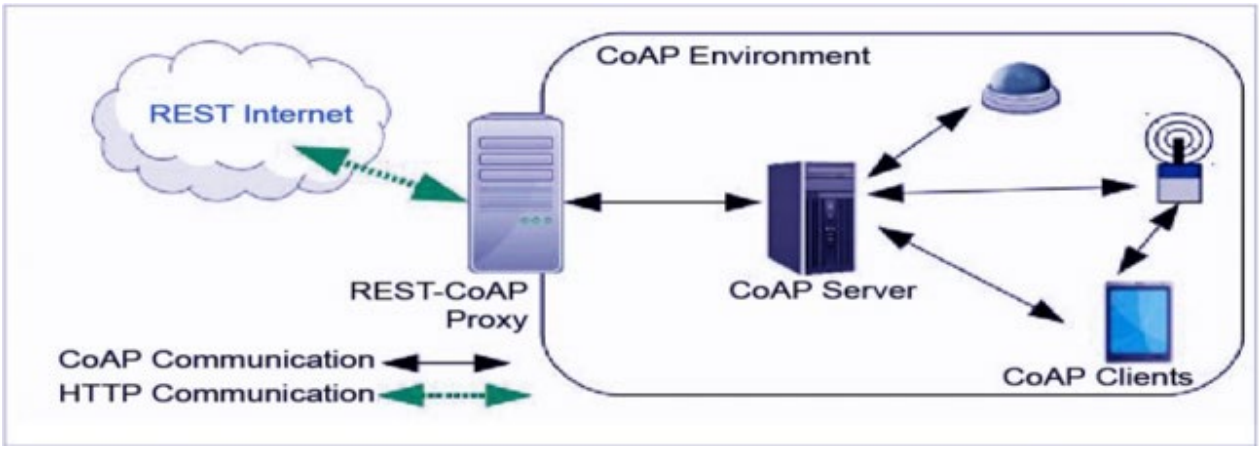

Figure 30: CoAP architecture (23)

CoAP is made of two layers: the request/ Response and message layers, as shown in Figure 31.

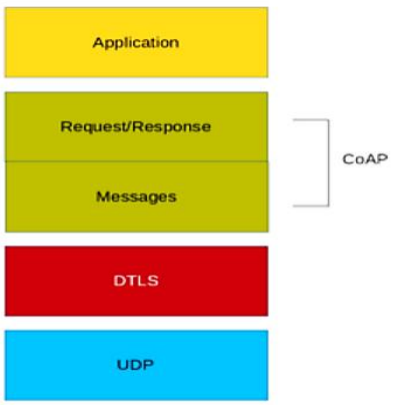

Figure 31: CoAP protocol layers (24)

Messages Layer deals with UDP messages exchanged between smart devices Each message is special, so each one will have a unique identification number for verifying whether it is already sent. CoAP Messages focuses on four different types of messages: confirmable, non-confirmable, acknowledgment and resets (82).

Request/response Layer depends on whether it is accurate or not with confirmable messages only or non. The request/answer layer is like REST with GET, PUT, POST and DELETE (24).

The two modes of communication in between devices and applications are reliable message transport and unreliable message transport. Reliability in CoAP is achieved by validated messages. Acknowledgment messages are distinct from confirmable messages. A Confirmable message will ensure that the target will receive it, and it will be repeated to the target before they accept it by sending an Ack.

Non-confirmable messages are inaccurate and are not suitable for use in applications that do not need confirmation. It is important to note that CoAP supports Piggyback, Separate Response, and Optional Relay communications (24).

CoAP protocol uses a secure data transfer protocol like the DTLS (Datagram Transport Layer Protection) over UDP (24). A recent improvement to TLS, because CoAP does not provide internal security by itself. In the CoAP variant of DTLS, it is called CoAPs. DTLS allows reliable communication between two computers. DTLS can implement AES, RSA. Thus, CoAP has four potential protection modes (83):

• Pre-shared password (PSK)
• Public/Private Key pair.
• No security: When DTLS has not been enabled.
• Certificate.

These protection modes are used to assess who can be trusted and what is trusted. Therefore, a solution needs a trusted authority or a certificate authority for use, as mentioned in IETF [27]. CoAP carries no security measure without security mode. A collection of keys will be pre-provisioned to create a DTLS session with both the Pre-shared key and raw public key.

Each application may use one key for a single device or one key for multiple devices, which provides an effective solution for symmetric encryption. The Raw Public Secret Key offers a DTLS Authenticate protocol, a validated public-key cryptography method. Also, devices have pre-configured keys that allow for the establishment of a DTLS session. Finally, in certificate mode, there should be a certificate anchor server. Each device that wants to initiate the DTLS session needs to have X.509 certificates and an asymmetric public key (83).

4.6 MQTT

Message Queuing Telemetry Transport protocol provided a structured messaging mechanism that was first implemented in the 1990s. MQTT is a reliable lightweight protocol that controls and tracks different physical sensors while remaining lightweight. MQTT operates by the client/server architecture and publishes/subscribe. MQTT is an effective protocol for restricted bandwidth, low power, insecure networks. MQTT is simple with a small header.

MQTT is composed of two components servers or brokers and clients or publishers. Clients may function in either subscriber or publisher mode, while brokers facilitate the contact between clients. Since subscribers are usually the publisher (sender), subscribers must subscribe to the posted subject by the publisher (sender) (24).

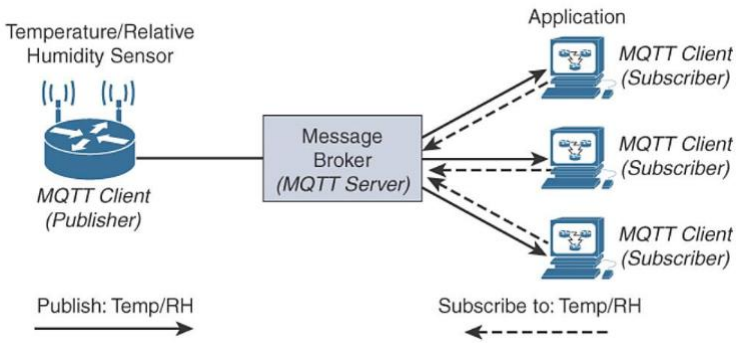

Figure 32: MQTT protocol framework (18)

Figure 32 illustrates, the MQTT client, like humidity and temperature sensors, publishes their data by transmitting it to the MQTT server, which acts as a message broker. So, the MQTT server accepts the publisher's data, and the client Sender may publish data. It manages subscriptions and exchanges data with an MQTT client, which acts as a subscriber. Subscribing is when a client will receive data from a broker, and publishing is when the client or computer will send data to the Broker.

In this method, the subscriber and publisher need not be simultaneously accessible because the Broker will buffer and cache information in connection failure. MQTT communication is done via TCP port 1883 between the server and computer. The messages exchanged between clients CONNECT/CONNACK, SUBSCRIBE/SUBACK, UNSUBSCRIBE/UNSUBACK, PUBLISH/PUBACK, PUBREC, PUBREL, PUBCOMP (23)

The client uses the CONNECT/ CONNACK to access the Broker remotely. The Broker allows SUBSCRIBE/ UNSUBSCRIBE to introduce new clients and delete clients via topics. Finally, PUBLISH/ PUBACK, PUBREC, PUBREL, PUBCOMP is used to send or receive electronic messages between a publisher and their client (23).

This protocol is very lightweight and defined by a 2-byte header accompanied by an optional header variable and a fixed-length payload. In MQTT, each session starts with client and server authentication, followed by data exchange. In IoT, instead of transmitting between devices through the HTTP protocol, they use MQTT.

There are three types of priority levels in MQTT. There is no guarantee that the message received in the lowest standard of service or additional packets are retransmitted. QoS 1 implies that the message is sent, but the recipient is not the sender. When receiving the QoS2 messages, the recipient will receive only one message (23). This message is then forwarded. When receiving the QoS2 messages, the recipient will receive only one message. This message is then forwarded[26].HTTP needs more overhead and is slower, but it uses less power because of its smaller packet size. In HTTP, each request requires a new connection, but with MQTT, the client and server will carry on after each message.

Even though it is lightweight, MQTT has a range of disadvantages. First, brokers must add support for TCP connections and communications, and those TCP connections must always be accessible. The topic names used to subscribe are more extended than allowed, which prevents the higher layers from compiling (23). MQTT-SN is another variant of MQTT called MQTT-SN, which is preferred over TCP because it uses UDP instead.

4.7 AMQP

Advanced Message Queuing Protocol is a more modern protocol based on a high-level message queue used in heterogeneous platforms; it started in 2003 by John O'Hara at JPMorgan Chase in London (84). AMQP demonstrates outstanding features by message queues, sending multi-function messages to queues in the system with adequate security. Large quantities of data can be sent securely to the data store without significantly affecting the whole system's performance. It is advantageous for the exchange of bulk messages as opposed to the RESTful Web services. AMQP is a protected messaging protocol used for message publishing and subscription. Customers go into different sized lineups for each service. Figure 18 is a description of the data transmission protocol with AMQP (25)

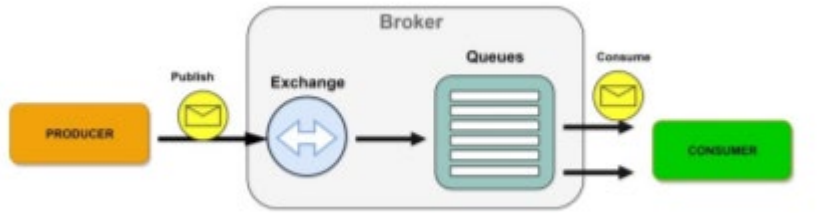

Figure 33: Overview of AMQP Protocol data transmission (25)

When the Producer publishes messages to the Broker, the Broker transfers the messages to the consumer and does not transfer them directly to the consumer. They will offer an Exchange instead. This exchange would assure reliable transmission of messages to their appropriate queues based on routing keys, binding keys, or the form of exchange. Before the consumer deletes them, the packets will be maintained in the queue. Using exchange between routing is the main difference between MQTT and AMQP (25) (84).

There is a much-used AMQP broker called RabbitMQ, which is common in the financial world. The RabbitMQ cloud system is convenient, easy to use and update, and available for Linux, Mac OS X, and Windows. AMQP middleware helps in the transmission of data no matter the language used. The standard assumes TCP as the transport protocol of choice for reliable transport. The messages in AMQP are clear, and the data inside each message is fixed and impenetrable. AMQP provides various ways to send messages such as point-to-point, publish-and-subscribe, and store-and-forward. An AMQP messaging framework consists of a variety of components (84):

- Publisher: The message producer will deliver messages to the AMQP broker.

- Consumer/Subscriber: Based on an application request, a Subscriber receives messages from one or more publishers via a broker.
- Broker/Server: A server or daemon application that monitors the number of virtual hosts, Bindings, Exchanges and includes several "virtual" messaging queues.
- Bindings: Binding is a connection between an exchange and a queue that uses a set of rules to route messages between the queue and the exchange.
- Virtual Host: A daemon mechanism in Unix-like operating systems that provides host, exchange, and messaging queues. Any of these systems run on their hosting tools or computers.

AMQP implements a messaging system that ensures high quality of service, asynchronous communication between two or more modules within an application, and communication between different application modules. AMQP offers a network communication protocol that defines messages transmitted between software systems—supplied by different vendors. The AMQP has features like queuing, routing, reliability, and security. Aside from providing a flexible architecture, AMQP implementations use high availability technologies to ensure the process continues even in failures.

	Transport protocol	Standard	QoS	Restful Support	Security
MQTT	TCP	OASIS	YES	NO	TLS/SSL
CoAP	UDP	IETF	NO	YES	DTLS
AMQP	TCP	OASIS&ISO	YES	YES	TLS/SASL

Table 5: Critical features of CoAP, MQTT and AMQP

The architecture plan has many parallels to MQTT. The Middleware(MOM) architecture's basic design is straightforward. Producers(client applications) create and send messages to an AMQP broker; deep inside the Broker, the messages are routed and filtered until they arrive at queues where another program called the consumers is linked. They then process and evaluate the messages (26).

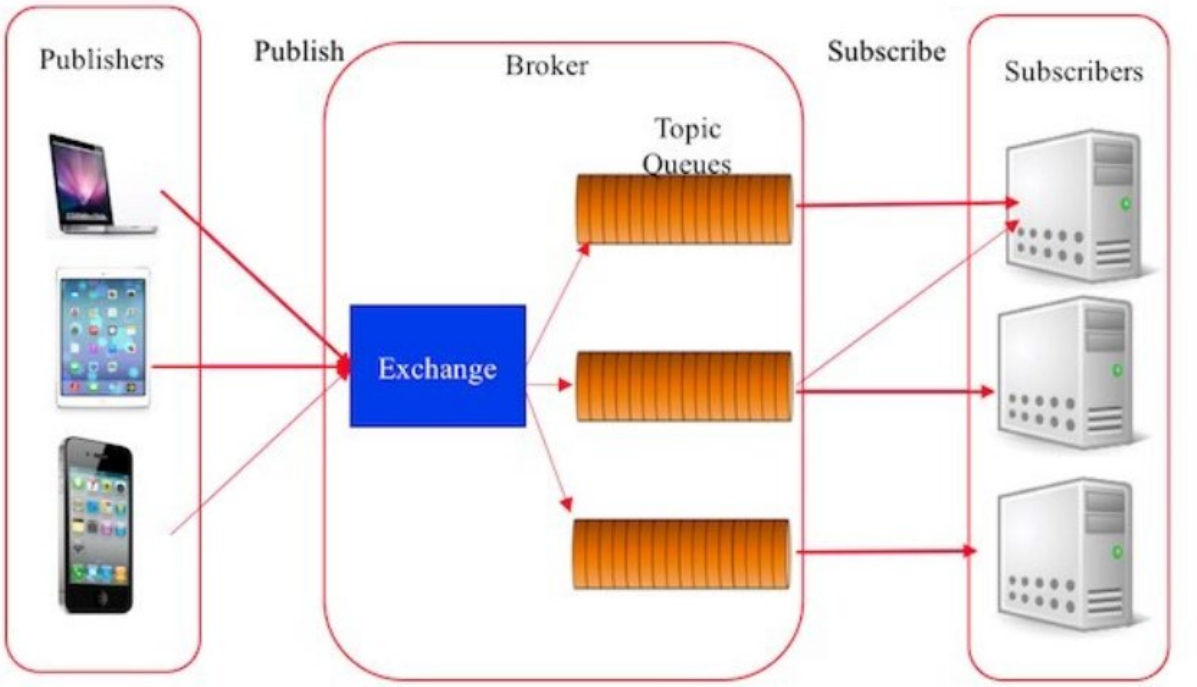

Figure 34:AMQP simplified architecture (26)

When an exchange has a message, it uses its routing key and three different methods to send messages, depending on which server is used by the person who sent it (26):

- Direct Exchange: The routing key matches the queue's name.
- Fanout Exchange: This message is duplicated and sent to all internal queues connected to this exchange.
- Topic Exchange: Any of the queues can handle the messages.

4.8 TR-069

TR-069 is also known as CPE WAN Management Protocol, based on XML/SOAP to ensure data communications between the CPE and ACS. TR-069 makes it possible to deploy IoT smart devices and offers virtualized services with multiple remote control and networking options. TR-069 ensures the secure and efficient sharing of data between devices in the Internet of Things network. A system for a CPE auto-configuration, incorporating CPE management functions into a standard framework (85). The TR-069 protocol is a proven and standardized technology, and most networks and service providers can quickly implement it today. TR-069 protocol standard was adopted by the Broadband Forum, an association that seeks to build and implement broadband networks (27).

- Auto-Configuration Server(ACS): A software that manages devices remotely.
- Customer Premises Equipment (CPE): CPE is commonly called a device like a Wi-Fi router that provides internet access via DSL(Digital Service Line), Cable, LTE, or WiMAX.

4.8.1 Functional Components:

The CPE WAN Management Protocol is dedicated to supporting a wide variety of functionalities that handle unique features of CPE, including the following (85)

Auto-Configuration:

TR-069 allows the ACS to provision a CPE or collection of CPEs at the time of first connection to the broadband network and the freedom to provision additional CPEs at a later date. This allows the provision of ACS-initiated re-provisioning of CPEs at any time.

The protocol's security mechanisms allow different CPE provisions based on specifications for each CPE or collective criteria such as CPE provider, model, software version, or other criteria. It provides optional tools for managing the CPE-specific components of optional services, such as payments, that require additional security monitoring.

The provisioning mechanism would allow the potential inclusion of new features, including new and enhanced features and functionality in future versions specification.

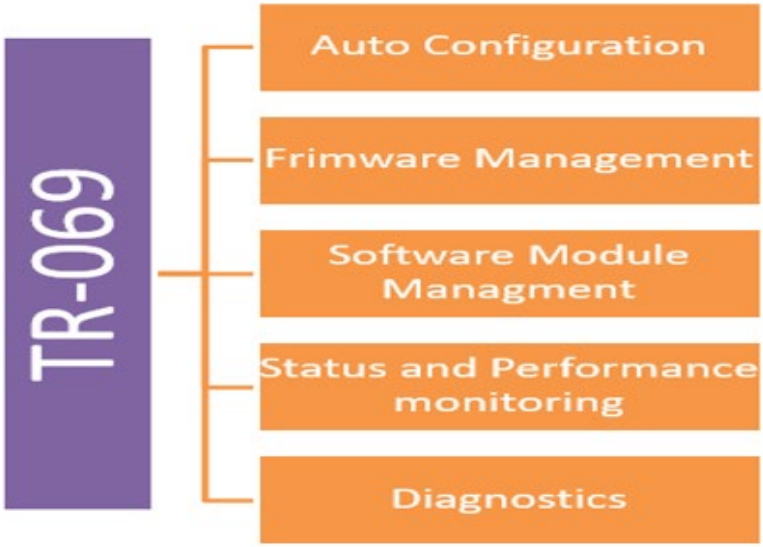

Figure 35: Functional Components of TR-069

Firmware Management:

Following the CPE WAN Management Protocol, tools allow downloading of CPE nodes like files containing compressed executables(software/firmware image files). The protocol offers mechanisms for version control. The ACS sends an electronic file transfer status update, monitors the download, and records the progress or failure.

Software Module Management :

The CPE WAN Management Protocol allows an administrator to handle the applications available with the CPE and its execution environments. The program can be mounted, modified, and uninstalled and notifies the ACS of each operation's outcome. The protocol also supports starting and stopping applications on the CPE, allowing, and disabling execution environments, and enumerating the software modules available on the system.

Status and Performance monitoring:

TR-069 helps the ACS obtain CPE usage information to track and improve(Monitor and Develop) the CPE performance. It explains how the CPE changes to state and notifies the ACS of its state changes.

Diagnostics:

It includes a function that enables a CPE to provide diagnostic information to the ACS, enabling it to diagnose and solve communication problems.

4.8.2 Connectivity between ACS and CPE

The relation between the ACS and the CPE is not a constant one. The unit can only attach to the ACS if/when need it. It usually lasts many seconds but should not be too long because it is used to exchange important messages. The brief exchange of messages is known as provisioning sessions (27).

The session is divided into several steps:

- The session is always initiated by the system that establishes the connection to the ACS.
- The ACS must verify the user's credentials (username and password) before starting the session. A password is submitted only if the digest-MD5 form is used. The HTTPS protocol can be more secure through the use of shared certificate authentication.
- Devices are detected by reacting with data from the provisioning session during initial session initialization. Most notably, a device's serial number is the primary identifier of an ACS device. For use later with the ACS GUI, a MAC address is not used to label the device but is saved by the ACS.
- Tasks execution on device, when the device is recognized and the contact component ends, the session's critical process begins, and ACS orders various tasks. It may involve reading and writing variables, diagnosing issues, rebooting, and moving data.
- The session is terminated once all the tasks have been completed. Anything extra should be started in a new session.

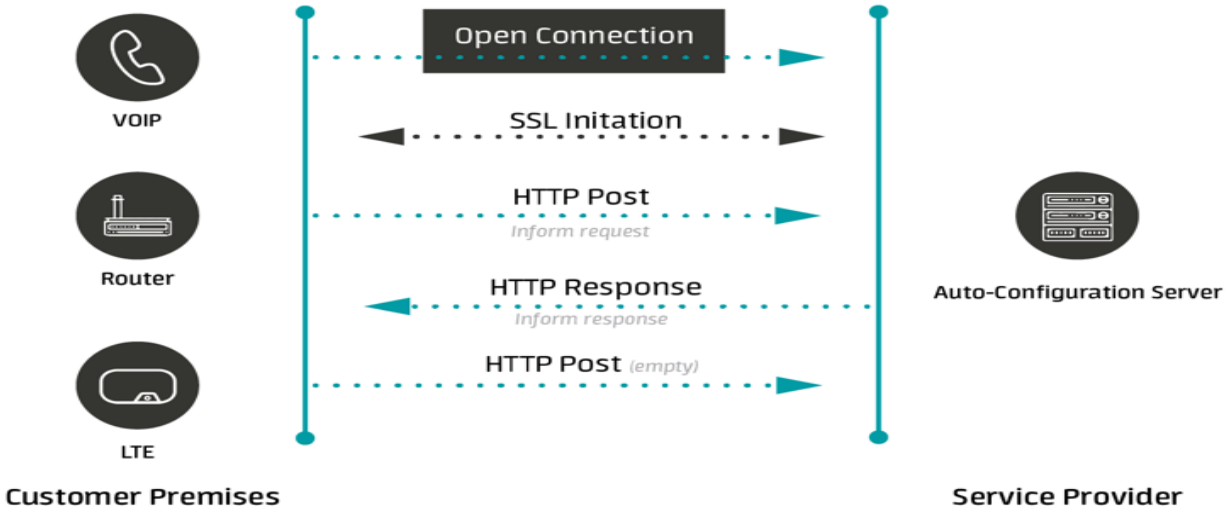

Figure 36: Session between ACS and End Devices (27)

4.8.3 Advantages of managing devices via TR-069

- It is easier to customize device settings than settings installed by default (27).
- It shortens our equipment installation time at the customer's premises by the machine automated configuration (27).
- It minimizes the number of engineer's visits using remote access. It involves turning services on or off and checking (27).
- It enables maintenance tasks, such as updating the firmware of a device and restoring its configuration. In addition to this, these operations may be planned for off-peak hours or periods (27).
- It improves performance across network optimization settings, like which wireless channels function best (27).
- It regulates the network condition by controlling the network (27).
- This software collects market analysis data, including identifying active users, such as customers whom other deals can make (27).

4.9 OMA-LWM2M

OMA-DM started a long time back, and its use in mobile devices. Hence, operators and enterprises use this to manage mobile devices remotely. Mobile phones that are managed with these protocols are configured on the network. LWM2M protocol ensures that M2M or IoT devices can communicate with each other and the broader network. LWM2M acts very well at incorporating IoT devices and allowing multiple vendors to co-exist in an IoT ecosystem. The data model can be generalized to run several industries (28).

LWM2M protocol design is familiar with element DM for mobile features in IoT networks. Some of the critical things for lightweight device management are suitability for the whole IoT market, not just for cellular but also for Wi-Fi devices, 6loWan sensor network devices just about anything that uses IP. It allows us to break down silos in M2M are traditional to make an interoperable device with cloud service and private service can manage.

LWM2M standard is not only for device management but also for application data, and that is something different from traditional device management solutions. They are used for managing the device but not for the applications. This standard includes an extensible object model and open registry for the actual semantics used, not just device management.

Bootstrapping interface is IoT headless device management to configure the devices with the exemplary service without pre-configuring it at the factory, which significantly reduces cost and optimizes time-to-market for the product or service. The device configuration function allows the provider to access device instances and resources, which enables to change the device settings and parameters instantly. The user can obtain error reports from devices when the service no longer works properly and send queries about device status using the Fault Management function. One of the most significant security risks on older embedded devices is due to lack of software updates; new risks that come with legacy software on the device can be avoided using remote firmware updates. These functions can be achieved using this lightweight M2M standard (86).

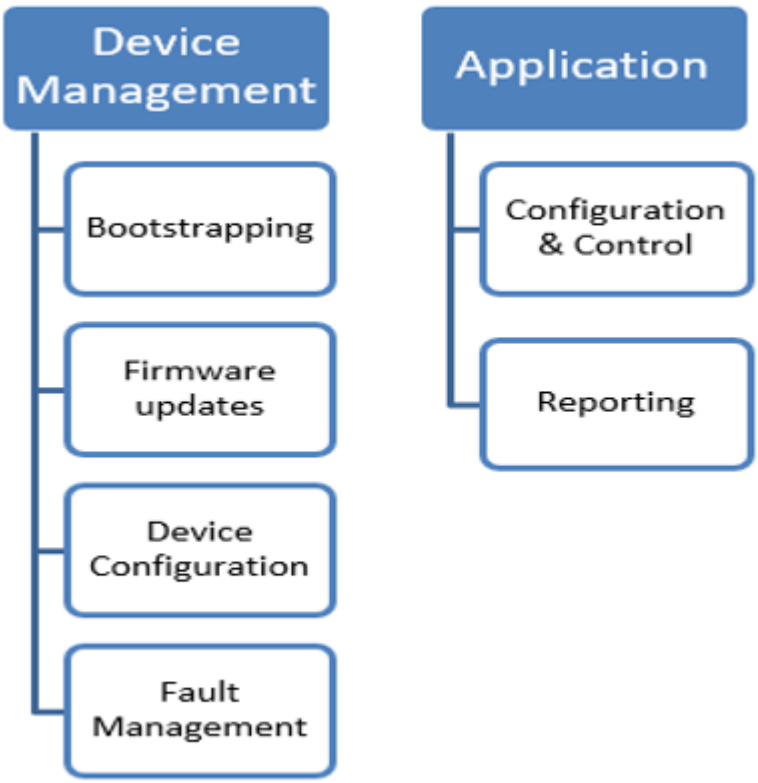

Figure 37: OMA-DM Protocol functions

Functional Components of lightweight M2M standard (28)

- Bootstrapping: Key management, Service Provisioning
- Firmware Updates: device update application and systems software with bug fixes.
- Device Configuration: To changes the settings and parameters of the device.
- Fault Management: Report Errors from devices and general queries about the device status.
- Configuration & Control: Settings of application configure and send control commands.
- Reporting: Notify changes in sensor values, alarms, and events.

4.9.1 LWM2M Architecture:

The protocol stack is simple. We have lightweight M2M, in which the functionality is built on top of collapse running over DTS UDP or an SMS bear. The object model defines resources that can be multiple instances of an object. Anyone can create an object with multiple instances and access to be serialized onto the actual protocol itself in a couple of different modes. The protocol itself is straightforward but also efficient. LWM2M in clients ranges from 5 kilobytes of flash to a kilobyte of RAM starting, and they can be tiny. The information reporting interface, which depends on CoAP technology, is used to get a long-lived notification style about a resource.

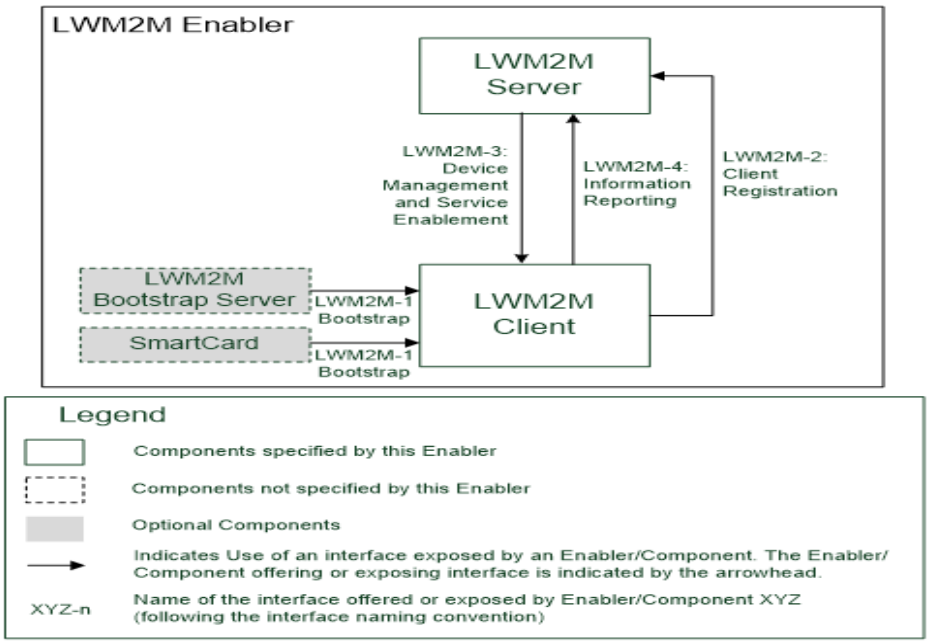

Figure 38: LWM2M Architectural Model (28)

- LWM2M server: Reuses IEFT technologies like CoAP protocol, DTLS, Resource Directory
- LWM2m Client(CoAP Servers): Device abstraction through CoAP, any IP network connections.

4.9.2 Benefits of LWM2M

The market needs something simple and efficient way for connectivity. For example, the Arm Cortex M0 microcontroller fits a thumbtack's tip and costs less than a dollar to produce. We need to manage the system because it is on the Internet, and its software can modify seamlessly. The Object Management Community has developed a global registry of computer elements. We can define semantics not only for device management but also for applications (86).

LWM2M standard also combines both the feet device management plane and the application plane. So traditionally, the device would have to keep up multiple stacks of protocols security. With this model, we can have one stack of technology and one library to do both. We have something that can be deployed now, the standard done, all the underlying technology. We can get many benefits that operators primarily have already understood from LWM2M device management. Two other standards already use IEFT standards, and it also fits into things like one under one such part of the standards ecosystem. A few of the noticeable advantages of LWM2M over other IoT Protocols are (87) (86):

- LWM2M is a reliable and straightforward security protocol for devices with limited resources.
- The encryption offered is based on the advanced DTLS protocol, which offers passwords based on pre-shared and public keys, certificates, confidentiality, implements authentication and data integrity.
- It provides a well-defined quality framework for hardware and software components that vendors can enforce to allow various secure system bootstrapping and device reporting.
- It offers a versatile, scalable, and vendor-agnostic solution for system management that improves from time to time, making it particularly appropriate for low-power devices with limited processing and storage capabilities.
- Considering all of this, LWM2M is the ideal option for IoT implementations involving various platforms and standards.

Chapter5

Machine Learning Applications in IoT

According to the IDC statistical report, over 50 billion IoT devices in the world will generate more than 60 ZB data (88). Complex systems can be built to improve the quality of life by collecting information from these IoT devices and analyzing them to sense and understand the environment, such as machine condition diagnosis, human body operations, health monitoring. Many scientists are studying machine learning to solve network issues such as traffic engineering, device identification, and security. This work focuses on providing an overview of IoT machine learning applications, highlighting recent advances in IoT machine learning techniques.

5.1 IoT Device Identification

Identification of devices is a process that predicts the form of an IoT system based on its characteristics. Understanding IoT device identification is critical for commercial service providers and security infrastructure managers to find vulnerable devices. Primarily, identify the IoT system recognition issue as follows: The input is separate information obtained from a computer; the input is different data gathered from a device, e.g., data from sensors, network data. The output is an IoT device label that indicates its type. Because of the proliferation of mobile computing, IoT deployment, this issue has received extensive attention in recent years (88). Table- 6 gives a review of the recent research work on IoT device identification. Researchers have switched to machine learning methods, which can also be passive identifiers since constructive approaches are focused on IP addresses; MAC addresses are not secure.

Identification of Generic IoT Devices:

A decision tree with random forestry and multi-classification is added to boost and recognize ZigBee devices. The proposed scheme initially defined radio signal's statistical characteristics, such as the immediate amplitude-phase and frequency spectrum (89). The system then gathered the features for available devices and trained models of the decision tree. Capture the device's features and enter the features into the classifiers for device identification. It validated the proposed scheme with Ninety percent accuracy in a given networked system; this scheme can also detect unknown ZigBee devices. One impressive future job is to improve the function space and verify whether accuracy can be enhanced (88).

SVM is for image-based identification of cameras. The proposed system leverages the detailed process of photo-taking of cameras and then deduces a camera's features. Specifically, more than 10,000 characteristics are used on the co-occurrence matrix, colour dependencies, and an image's conditional probability (90). To classify different images for different cameras, the system then trains an SVM model. On a public image database, the SVM scheme validated the suggested scheme. A radial basis function (RBF) based SVM was trained on 100 images data set and tested the same set later. Experimental findings have shown that the precision of identification reaches more than 97 percent (88).

Technique	Device	Accuracy	Work
Decision tree, ensemble learning	ZigBee device	90%	(89)
KNN, GMM	Android, iOS	98%	(91)
KNN, SVM	Android	90%	(92)
Random forestry	Android, iOS	95%	(93)
SVM	Camera	97%	(90)

Table 6: Latest research work conducted on IoT device identification

Identification of mobile devices:

Based on network contact traffic, KNN and SVM are used to classify mobile phones. The suggested system records mobile traffic for a period using TCP dump and further transforms the collected data into a 23-entry function. Based on traffic bursting patterns, the function is heuristically specified and not linked to explicit payload content. The collected data is then educated using SVM and KNN to classify mobile phones and applied for the qualified classifier for potential phone recognition. SVM and KNN evaluated the suggested scheme on 20 Android mobile devices operating 3G touch connections. The accuracy approaches 90 percent. The measurement is also efficient: about 15 minutes is the time required to locate a cell phone. The findings suggest that cell phones can be monitored accurately, even though the contact traffic is encrypted (92).

KNN and GMM use acoustic data to track cell phones. The computer captured and collected the acoustic signal features from the microphone. The ML models were then conditioned on the data captured in the proposed methodology. Later, to classify cell phones, the KNN is used. The proposed scheme has been tested on 52 mobile phones in the lab. There are both iOS and Android

systems (93). Research indicates that it was possible to classify devices produced by various vendors efficiently. Also, it is possible to classify devices from the same maker and model. The precision is as high as 98% (88).

5.2 Security

Security concerns in IoT networks are becoming more prevalent. Critical with the increasing number of attacks nowadays. Security concerns in IoT networks are becoming more prevalent because of IoT devices and communication protocols. For example, IoT devices are usually equipped with a lower battery and micro-controller. Thus it is easy to be flooded; IoT communicates with each other, which are more vulnerable to attacks via Bluetooth, ZigBee, WIFI or GSM. There are typically three elements in an IoT network, including devices, gateways, and controllers. These components could become targets of potential attackers. In the model, traffic, wireless signals, device events and configuration files could be analyzed. Different ML methods for classifying the data are used with the function extracted from the data sources. The observations can be used for protection and authentication and assess whether the incident is interference or an exception (88), where multilayered perceptron and probabilistic neural network combinations were used.

Technique	Problem	Accuracy	Work
Combined ANN	Security of an IoT element	100%	(94)
KNN/SVM/decision tree	Authentication of an IoT element	80%	(95)
Bayes Algorithm	Security of an IoT element	None	(96)
SVM	Intrusion detection	100%	(97)
SVM/NN	Security of Mobile networks	None	(98)

Table 7: Latest research work conducted on IoT Security

The paper used the amount of traffic to forecast the state of an IoT item, the rate of operation, the rate of losses, the queue length of packets, and user intervention for the inquiry as indicators to clearly define the state of an element (94). It is anticipated that the solution would reduce the cost of administration, especially during an emergency. In MathCAD, the suggested solution was assessed. The findings show that the combined ANN model can give higher precision. The combined model convergence was close to 100 percent relative to the recurrent ANN model, wherein in some cases, the recurring model fell to 15 percent (88).

RF fingerprinting is used for system authentication, measured for cellular communication devices, including GSM, WIFI based on dispersion and permutation entropy (95). It is not hard to bypass the technique since it is dependent on the cellular device's external characteristics. The paper's authors compared various ML classification strategies, including decision tree, SVM, and KNN. The authors used a series of wireless devices with nine nRF24LU1+, where the RF signals are transmitted. Software-defined radio was able to capture the signals. The findings show that overall accuracy can exceed 80 percent, enough to support wireless IoT devices that support multi-authentication (88).

For IoT devices, a general security framework is generated. We do not currently own a unique protection mechanism that adapts to IoT devices due to IoT device's growing use. This paper's structure classifies the devices into various categories, indicating the ability to support protection mechanisms based on their capabilities and parameters, such as control, encoding, and scalability and network layer. For the above reason, the authors also found that the naive Bayes algorithm is suitable. Therefore, after entering a file, the file consists of specified assets. The system outputs the unit's class, e.g., Class A for Critical, Class B for Medium, and Class C for Non-Critical (96).

For the smart home, called IoT-IDM, an intrusion detection and mitigation device is proposed. The proposed system gathers information on traffic and activities from different devices in a smart home. The information will be transferred to the controller of the SDN, where IoT-IDM is deployed. Finally, the linear regression model and SVMs are used by IoT-IDM to achieve the optimum classification model. With an experimental setup where the Philips lighting system is used, and a realistic environment is used, the proposed scheme was tested. The linear regression model's accuracy was 96.2 percent, and the model SVMs accuracy was 100 percent (97).

A mobile protection framework using machine learning was proposed by Do et al.[47]. The new framework will strengthen the vulnerabilities of mobile networks. The machine learning-based framework can help solve security problems, including zero-day attacks and the creation of definitive threat signatures, compared with previous systems. The authors also submitted a case study with the IMSI catcher about the Man-In-The-Middle attack. In the case study, to classify the

anomalies, SVN and neural networks are used (98).

5.3 Traffic Profiling

Traffic profiling refers to characterizing traffic patterns of communication networks, including IP, wireless and mobile networks. It provides insightful traffic data, thus managing and engineering the network for better performance. For example, detecting abnormal traffic increases the safety of underlying networks that have made significant research efforts in recent years.

A traffic profiling issue is defined as traffic profiling task input is collected accurate packet forwarding data; the outcome is a compilation of traffic-based variations. Typically, studies focus on network traffic statistical properties. This method, while obtaining valuable engineering network data, is limited to specific networks. In current days research teams have exploited the ML power to profile network traffic, achieving broader results.

Technique	Problem	Application	Work
Clustering	Traffic pattern analysis	Traffic analysis	(99)
Frequent itemset mining	P2P identification	App. Identification	(100)
Frequent item set mining	Anomaly detection	Intrusion detection	(101)
K means	Network user profiling	User profiling	(102)
ML algorithm set	Traffic classification	App. Identification	(103)

Table 8: Latest research work conducted on Traffic Profiling

The clustering technique was used to profile IP network traffic. First, the scheme captures traffic data and aggregates information into streams. Each flow has five dimensions, i.e. the source IP, the destination IP, the source port, the destination port, the protocol type. Next, for each dimension, the scheme clusters information, i.e. the source's IP dimension, the destination IP dimension Outputs are meaningful clusters. The IP address data reveals the node's traffic patterns; the port data shows the service's patterns. They both include basic traffic patterns. In the scheme, a newly proposed entropy-based metric was used to determine how many clusters are produced. The

scheme then analyzed the structures, i.e. resemblances and difference, of the traffic of each cluster. The system also studies how the observed structure evolves. The system employed dominant state analysis to model each cluster's five-dimensional interaction based on the structures found (99). Frequent item-set mining is used to detect network traffic anomalies. Seven-entry tuples, i.e., srcIP, dstIP, srcPort, dstPort, protocol, #packets, #bytes of captured traffic, are processed in the proposed method. The system first employs a traditional histogram-based detector to filter out suspicious flows. For the filtered traffic, the scheme establishes a transaction with seven items for each suspicious flow. Then, the system uses frequent object set mining to find the anomalies.

For instance, when an IP address is flagged as a frequent element collection, it may be an exception. The choice of everyday products is the execution of the proposed method. The scheme was validated on a median ISP. Next, the ground truth is detected using a manual analysis based on top-k queries. Then, for anomalies to be identified, the system was used. There is the lowest possible number of false alarms in the proposed scheme (100). The most notable gain of this strategy is that it decreases the time taken to analyze deviations as observed. One challenging aspect of this approach is parameter selection. In its current form, the criteria for frequent object set mining is by trial and error in its current state (100).

Frequent itemset mining is employed to identify P2P traffic from multiple network traffic. The proposed scheme's main idea is to extract dominant, unique features from P2P traffic using frequent items and mining (101). The device first tracks P2P traffic sensitively as qualified data to acquire the characteristics. The knowledge is translated into regular five-tuple structures, i.e., source IP, destination IP, destination and source ports, protocol type, and manually used statistical communication flow properties.

The scheme then uses frequent items and mining to find patterns above the threshold. Later, patterns are used to identify P2P traffic for online network traffic. Note that many engineering, scientific, and heuristic efforts are being made to achieve good results. During the scheme's proposed performance evaluation, 10–15 patterns are obtained for BitTorrent traffic captured in the campus network. These extracted patterns show more than 90 percent accuracy when tested on real traffic. This approach to other approaches is an interesting future work along these lines (101).

K-means variant is used for network context information that is centered on internet activity.

The suggested scheme was planned to accommodate DNS requests. The information is a vector that indicates how many times a flow accesses a given domain name; the cumulative number of different domains is the vector size. The scheme then uses a modified K-means clustering algorithm for different group flows, which denotes all specific user traffic. The scheme has thus successfully identified users of the Internet. The method is validated on a DNS server on the campus network. In two months, as many as 19 percent of users have been fully identified; 73 percent of users in this subgroup can be logged in over 56 days (102). The scheme is beneficial, and it does not rely on conventional IP and port knowledge. It would be essential to see how this approach is extended to other network traffic forms (88).

Naive Bayes decision tree is used to categorize traffic on the network in specific high applications. The proposed scheme set out several facts and figures on the initial negotiation round of the topmost application. The machine then qualified various classifiers for the traffic with these statistics as characteristics and well-captured information from flows.

The well-trained classifiers are later used for future traffic detection. The suggested strategy was tested on traffic dynamics on campus. Experimental results show that classifiers with the newly defined statistics have an average of 92% accuracy. Compared to the same classifier without the newly defined function, the accuracy improves by about 7% (103). To use this tool, one must first evaluate the total number of application types; its usefulness for unknown traffics is unknown.

5.4 Industrial applications

Firstly, since the early eighties, commercial and industrial applications have been successful. These implementations are, for the most part, based on an ML subarea called learning from examples. ML algorithms predominantly handle simulated data, while real-world problems generally require large-scale computational datasets to be handled. Research in constructive induction and learning relationships is connected to integrate domain-specific properties into the acquired knowledge, which has not been adequately considered in ML research. To tackle the applications of the real world, ML should combine many other approaches; Table 9 offers an overview of recent research work.

Technique	Device	Accuracy	Work
Bayesian network, naive Bayes, decision tree, random forest	Smart meter operation	96.90%	(104)
Clustering algorithm	Parking space detection	97%	(105)
Hidden Markov model	Grape disease prediction	90.90%	(106)
SVM	Flowering dynamics in rice	80%	(107)

Table 9: Latest research work conducted on Industrial Applications

(104) has used ML to increase the performance of the operation of smart meters. With the enormous rising number of smart meters, administrators need to guarantee the cost-efficiency of their operations. In this paper, the authors used different ML methods to predict whether administrators should send a technician to a client venue. The system will reduce a lot of travel costs and human capital with greater precision. Testing of the models was carried out using data from a commercial network. Different classification algorithms have been tested, including Bayesian network, naive Bayes, decision tree and random forest. Finally, the findings indicate that random forests achieve the highest precision, which is 96.69 percent, and the estimated cost of saving for the commercial network is around $1 million US.

(105) has developed an IoT-based system that automatically detects parking space occupancy. The machine uploads the pictures collected first. A vehicle recognition feature is then used to discover the parking spot. After that, to identify the most commonly parked sites, a feature clustering algorithm based on Mean-shift is used. The authors are checking the device on a model Raspberry Pi 3. Camera information is gathered for restoration and observation on a local street near the University of Washington campus linked to AWS IoT. The findings indicate that 97 percent can be done with real-time precision.

(106) proposed an agricultural system capable of monitoring the environmental conditions of the vineyard and predicting early-stage grape diseases. To track temperatures, moisture and humidity throughout the yards, the device used a wide range of sensors. The data will be transmitted to servers using ZigBee, where proposed an agricultural system will implement a hidden Markov

model. Each state represents a specific condition within the hidden Markov model. Since November 2015, the author has been implementing the system in the real world. The findings show that the hidden Markov model's precision is 90.9 percent, significantly enhancing the statistical method's accuracy.

(107) suggested an innovative method to detect the characterization of the dynamics of rice flowering. The method first collects time series from the rice fields with photos. Secondly, from the images, the process extracts local feature points. The technique will produce visual words as the object-recognition strategy during the third stage. SVM is used to classify the time series of images and detect the flowering component. During the different periods with different rice varieties, the authors collected image data for evaluation. The findings show that for counting many flowering panicles, the technique performs well. When selecting accurate training data, the accuracy of classification can be more than 80 percent.

5.5 Deep learning application in IoT

The forecast's accuracy can be determined by Deep Learning (DL), a new ML breed. Because of their self-service design, deep learning models are ideally adapted to classification and prediction tasks in creative IoT frameworks that include contextual and customized assistance.

5.5.1 Health Care

In transforming health care, acquiring information and personalized recommendations from involved, complex biomedical data remains a crucial challenge. IoT, combined with deep learning, has now been used in providing individuals and communities with health and well-being solutions, and deep learning approaches could be the vehicle for translating extensive biomedical data into improved human health. Table 10 contains some typical innovative healthcare applications based on deep learning.

Category	Technique	Application	Work
Disease Analysis	CNN	Diabetic retinopathy detection in retinal fundus images	(108)
	CNN	Knee cartilage segmentation	(109)
	CNN	Smart personal health advisor	(110)
Health Monitoring	CNN+LSTM	Human activity recognition	(111)
	CNN	Energy expenditure estimation	(112)
	CNN	Arrhythmia detection and classification	(113)
Miscellaneous	Faster R-CNN	Medicine recognition	(114)
	CNN	Pill image Recognition	(115)
	CNN	Skin lesion classification	(116)

Table 10: Latest research work conducted in Health care

5.5.1.1 DISEASE ANALYSIS

An essential subject in healthcare is the classification and analysis of medical images. DL has been extensively used in supporting illnesses imaging techniques following computer vision success (110) CNNs have been used to deduce a classification of low-field knee Magnetic resonance scans to the cartilage segment instantaneously predict the risk of osteoarthritis (109) Another study (108) uses CNNs in retinal fundus photographs to identify diabetic retinopathy, achieving high

sensitivity and specificity in approximately 10,000 test images for certified ophthalmologist annotations. Some software focused on medical imaging techniques have also used DL. For example, (116) presents a pill image recognition model based on DL that uses smartphones to identify unknown prescription pills. (115) suggests a DL approach to classifying a thermotropic image containing a skin lesion as malignant or benign. An intelligent DL-based medicine recognition system called ST-Med-Box is being proposed by (114). ST-Med-Box can help chronic patients take multiple medications correctly and avoid taking incorrect medications, medication reminders on time, medication information, and chronic patient information management.

5.5.1.2 HEALTH MONITORING

Sensor-equipped smartphones and wearables nowadays typically allow different mobile health monitoring applications. However, the massive raw data call for a more efficient identification extraction model concealed the underlying representative characteristics. A promising opportunity for this problem is the implementation of the advancement of deep learning in activity recognition. (111) To analyze the movement data, create CNNs and LSTM and better combine the results to predict freezing gaits in patients with Parkinson's disease.

(112) Data from triaxial accelerometers and heart rate sensors are used to obtain promising results in predicting energy expenditure (EE) with the CNN model, helping relieve chronic diseases. The 34-layer convolutionary neural network is trained to map the ECG sample’s sequence to a rhythm class sequence. To detect a different range of myocardial infarctions from ECG recorded with a single-lead smart monitor, productivity outperformed physicians (113).

5.5.2 Smart Home

A smart home enables the interconnection of ubiquitous smart home devices and technical convergence and services through home networking to achieve a better living quality. In recent years, many systems have been developed to apply deep learning techniques in various smart home applications, as summarized in Table 11.

Category	Technique	Application	Work
Home Robotics	CNN	Autonomous navigation	(117)
	CNN	Hand-eye coordination for robot grasping	(118)
Indoor Localization	DRL	Bluetooth low energy signal strength based indoor localization	(119)
	RBM	CSI-based fingerprinting for indoor localization	(120)
	DNN	Semi-supervised Wi-Fi based localization	(121)

Table 11: Latest research work conducted on Smart Homes

5.5.2.1 Home Robotics

Home robots can perform different tasks in home environments with sensors, actuators and databases equipped. Home service robots should have critical functionalities, including location, navigation, construction mapping, the interaction between humans and robots, object recognition, and handling of objects. Case-specific approaches are required for robotic navigation in GPS-denied environments to control a mobile robot to any desired destination. A new approach to autonomous navigation is introduced using pattern recognition and DL techniques such as CNN to identify markers or objects from images and videos using the RGB-depth camera.

Computer intelligence techniques are implemented in the robot operating system and positioning of objects (117). On a screen installed on the assisted robot, multiple potential matching objects identified by the robot with a deep neural network object detector are displayed to enhance and evaluate human-robot interaction (HRI). An extensive convolutionary neural network to predict the probability that the gripper's task-space motion results in successful grips using monocular

camera images only, regardless of the camera calibration or the current robot pose improves the hand-eye coordination for the handling of the object (118).

5.5.2.2 INDOOR LOCALIZATION

With mobile device's proliferation, indoor localization gradually becomes a critical research issue since it is not viable to employ GPS in indoor environments. Indoor localization enables numerous smart home services, such as wireless intruder detection, elder monitoring, and baby monitoring. Nevertheless, it faces many propagation challenges like multi-path effect, fading, and delay distortion. High accuracy and short processing time are indispensable performance indicators while designing an indoor localization system.

Fingerprinting-based indoor localization is an effective method to satisfy the above requirements. Received Signal Strength Indication build on biometrics are fragile and incorrect. (121) propose a new SDELM algorithm that takes the lead in developing semi-supervised, deep learning and extreme learning machines. This approach achieves reasonable localization efficiency and, with full utilization of unstructured data, reduces the measurement effort.

(119) A model based on low energy like Bluetooth signal strengths based on semi-monitored DRL is proposed to generalize optimal policies as the inference mechanism; this technique incorporates variational autoencoders. (120) To obtain the locations, 4-layer RBMs are used to process the raw CSI information. However, the proposed system considers a device-oriented approach that would not work if individuals did not have cell phones or refused to connect their phones to APs.

5.5.3 Smart Transportation

"Intelligent Transportation Systems (ITS) apply a range of technology to track, analyze, and control transportation systems to increase performance and safety, according to the US Department of Transportation." Management, quality, and protection should all be present in smart transportation. In other words, smart transport uses modern and evolving technologies to make it faster, more cost-effective for both the community and the user, and cleaner to get through a city. These new opportunities are facilitated by digital developments, especially by the proliferation of IoT devices and 5G networking technology.

The former presents affordable sensors and controls built into virtually any physical system that can be remotely operated and handled. The latter provides the high-speed communications needed for real-time, low-latency management and control of transportation networks. Smart transit is not just a theory for the future; it is now being deployed in various cities, with the lessons learned from their experiences and shortcomings being extended to new systems. Table 12 contains some smart transportation applications based on deep learning.

Category	Technique	Application	Work
Autonomous Driving	CNN	End-to-end learning for self-driving cars	(122)
	FCN-LSTM	Learning driving from video datasets	(123)
	CNN	Real-time object detection for autonomous driving	(124)
Traffic Monitoring	CNN	object tracking	(125)
	RNN	object tracking	(126), (127)
	SDAE	Road accident detection	(128)
Traffic Prediction	SAE	Traffic Flow Prediction	(129)
	LSTM	Short term traffic prediction	(130)
	CNN	Crowd Flow prediction	(131)

Table 12: Latest research work conducted in Smart Transportation

5.5.3.1 Autonomous Driving:

A crucial component of city automation is computerized driving. There are two primary concepts for eyesight-build autonomous driving technologies: the facilitated approaches to perception and behavior reflex approaches. Autonomous driving systems currently focus more on real-time inference velocity, small model size, and energy efficiency (124). In order to learn a map from input images to driving behaviors, these self-driving driving video train systems build a direct map from sensory input to driving action.

In (123), the authors prepare a CNN to map unprocessed pixel value explicitly to the steering commands from a singular front-facing webcam. The authors in (122)put forth a programming method motivated by language models that trains an end-to-end FCN-LSTM infrastructure to anticipate the discrete and continuous number of co-driving behaviors. The system learns from the Long-term Convolutionary Recurring Network and extracts the video's spatial and temporal connections.

5.5.3.2 TRAFFIC MONITORING

The development of automated traffic monitoring systems, which play an essential role in reducing human operator's workload and warning drivers of dangerous situations, is one of the most attractive research fields in smart transportation. Video analytics of traffic has become an essential component of smart traffic monitoring systems. The following presents how deep learning is applied from three perspectives to traffic video analytics: object detection, object tracking, and face recognition.

In different scenarios, such as pedestrian detection, on-road vehicle detection, and unattended object detection, object detection has been implemented. To present a useful and accurate algorithm for monitoring using a single CNN for learning useful feature representations of the target object, it is essential to automatically monitor suspected individuals or target vehicles for safety monitoring, urban traffic management, and autonomous driving (125).

Where recurrent neural networks (RNN) are used, the end-to-end object tracking method has been proposed to map directly from unfiltered sensors input to object tracks in detector space without necessitating any platform identification (126), (127). (128) propose a system for

automatically detecting road accidents in surveillance videos using a stacked denoising autoencoder to learn how to view Spatio-temporal volumes of raw pixel amplitude instead of conventional hand-crafted characteristics.

5.5.3.3 Traffic Prediction

Traffic flow prediction is a problem in transportation modeling and management and intelligent transportation system design, which today relies heavily on historical and real-time traffic data collected from all sensors. To effectively use such massive heterogeneous data, conventional ML techniques, such as SVM, would consume much time and power-consuming computing resources. Besides, hand-engineered characteristics are insufficient to satisfy accuracy because of the restriction of related prior knowledge.

In (131), the authors propose an online SVR method to predict short traffic flows under typical and atypical conditions. There is a need to create several SVM models, and many memory resources need to be consumed. Recently, deep learning has attracted significant attention from academia and industry because of its capacity to extract inherent characteristics from data and exploit rich traffic data. These features are combined with the concept of multi-task learning from related roads and stations to explore the nature of the entire road transport network and predict traffic flow (129). The SAE (Stack of Autoencoders) model is proposed to extract historical data characteristics to predict these characteristics. Many works have focused on the prediction of traffic and crowd flow by using DL (130).

5.5.4 Industrial applications

Deep Learning Applications have made strides in addressing automated data pattern analysis, which has reached human beings. The drawbacks of standard machine learning algorithms have been overcome by deep learning in the last five years. It has caught the attention of corporations during its growth time, and everyone has an urge to make use of it. However, it is not easy to investigate and decide where to start and apply a deep learning model to solve their challenges. DL is one of the vital productions in hospitality, digital assistants and automotive operations. With the accelerated use of ML, organizations are leveraging their technologies to be part of Business 4.0. Table 13 summarized the recent research work conducted in Industrial applications.

Category	Technique	Application	Work
Fault Assessment	Wavelet-based CNN	Fault diagnosis and identification	(132)
Manufacture Inspection	CSAE	Transformer fault diagnosis	(133)
	CNN	Robust inspection system	(134)
	CNN	Surface integration inspection	(135)

Table 13: Latest research work conducted in Industrial Applications

5.5.4.1 FAULT ASSESSMENT

A smart factory must track machinery conditions to enforce smart manufacturing, detect incipient faults, diagnose the root cause of failures, and then integrate the knowledge into production and manufacturing control. For automated machinery fault diagnosis, a wavelet-based CNN is suggested in (132). A one-dimensional waveform is converted into a two-dimensional output that is fed into CNN using the wavelet transform.

5.5.4.2 MANUFACTURE INSPECTION

Various visual inspection methods have been proposed to derive representative features with specialist experience to identify product defects in large-scale manufacturing to examine and determine product output (133) reliably. Many of these approaches are focused on conventional machine learning strategies. Deep learning has recently risen to popularity as an effective visual inspection method. In their paper (134), the authors suggest a deep-learning-based classification model for implementing a consistent inspection framework. The fog computing environment is tailored to a CNN-based framework, increasing the system's computing performance. In (135), a standardized CNN-based method is suggested to remove patch features and predict defect areas for surface integration inspection tasks by thresholding and segmenting.

5.5.5 CHALLENGES

Machine learning, as seen in recent literature, produces satisfactory outcomes. That being said, there are several issues to be resolved for using machine learning in IoT applications.

DATA COLLECTION

The effectiveness of deep learning techniques depends on data sources. Even if the model's architecture is well built, the deep model cannot play a role if there is not enough clean data. Therefore, a crucial analysis dilemma is how to implement the data collection equipment. The number of sensors used and how may implement the sensor's effect on captured data accuracy. The secret to solving issues is the knowledge found in the results. For the entire IoT framework workflow, it is essential to build a data collection module. For example, developed DeepCham, a picture selection module, to boost the model's recognition accuracy. The data collection module integrates the principle of crowdsourcing. A cost-effective, efficient, and trustworthy data collection model is essential to build deep functional learning-based IoT applications.

MODEL TRAINING

Training a deep network involves cumbersome assignments. The depths assess, as we know, the ability of a deep learning network to retrieve essential characteristics. However, as deeper models develop, the gradient disappearing issue emerges, which deteriorates the output. To that end, we have built a system for pre-training models that includes stacking RBMs. The gradient vanishment issue may also be mitigated using the ReLU function to replace the sigmoid function. Overfitting is another significant issue we are faced in in-depth model training. The integral approach is to enter more data or to decrease the model parameters.

One powerful approach to minimize the number of parameters is to use convolutionary kernels, and utilizing the dropout is also an option. In recent years, CNN has made a significant breakthrough, with the number of layers in CNN models growing from five to over 200. Using the deep learning algorithm to deal with wireless network field problems, the methods described in these classical convolutional neural networks might be accurate. Most works utilize organized data sources like sensors, photographs, and documents. In the actual IoT universe, though, more data occurs in an unstructured fashion. Using machine learning based on these unstructured data is worth further study.

HARDWARE LIMITATION

Deep learning is a powerful tool for processing big data, resulting in its high hardware requirements. How to implement a deep model of a resource-limited embedded device is still a challenge. So far, there are two types of research aiming to solve the problem. One is only to treat end devices (like a smartphone) as data collectors. All data are transferred to resourceful servers to be analyzed. However, in this process, we may incur data disclosure, network failure issues. An alternative solution is to reduce the complexity of the networks with slight performance degradation.

Most IoT devices have smaller batteries and microcontrollers; battery and processing capacity restrict even the gateway. Some machine learning approaches, e.g., DNN, require loads of computational energy and are power-hungry. Therefore, spreading tasks to various computer nodes, saving power and computational resources, and achieving near-optimal accuracy are essential for IoT network applications.

SYSTEM DESIGN

A movement is developing to build a mobile-edge learning infrastructure that covers edge devices and the cloud. A cloud-edge framework may leverage the edge to minimize latency, boost safety and protection, and incorporate intelligent data preservation strategies. Besides, it will utilize the cloud to exchange knowledge through edge nodes, train specialized computational-intensive models, and make high-quality decisions. Some reports on the convergence of deep learning and edge computing have recently been published. In terms of resource capabilities, edge systems may be incredibly heterogeneous.

Security

Online real-time tracking sensors require online estimation and real-time feedback in many IoT applications, including health and industrial monitoring. The need for enhanced security, QoS, and machine sophistication is higher than in other applications. Security must address these issues in the future (88).

DDoS Assault Unlike the previous method of attack protection, a machine-based learning framework needs more computing resources. The choke-point becomes vulnerable to DDoS

assaults. Trading between precision and computer complexity is a vital direction for machine-learning-based security frameworks. Research on detecting security-based attacks and critical distribution channels has plenty of room (88).

Understand various methods of machine learning Another fascinating topic is how to choose machine-learning algorithms. Researchers also suggested several algorithms. Why various algorithms affect IoT system recognition and machine learning algorithms are yet to be learned. Privacy assessment Furthermore, all the analyses checked here demonstrates that IoT sensors can be exceptionally reliable. An in-depth assessment of privacy and possible protective methods are worth researching.

CONCLUSION

This work investigated how machine learning and deep learning bring new opportunities to the IoT. Many IoT applications have been empowered with these tools. For several pattern recognition and classification activities in the IoT, DL offers an ideal solution because it can comprehend hierarchical representations from the input data. With image as well as voice information, CNN operates splendidly fine. To forecast time series, the RNN, as well as the LSTM, are used. For dimensionality, AEs are used for High-Dimensional Data Reduction. For noisy situations, GANs are sufficient. In an unsupervised way, the RBM and indeed the DBN grab dynamic representations of data. However, DL models perform better than traditional machine learning processes, but the system's computational complexity impairs the DL model's implication in time-strict IoT applications. With deep learning, it is unnecessary to make efforts to design complex features. Deep learning offers us a new perspective to solve traditional problems and reveal new insights on IoT. However, designing architecture with high accuracy and low resource consumption is still an open issue. Adopting ML in IoT may solve problems at hand, but it opens new challenges. There is a lack of current machine learning literature on its uses for IoT programs and services, following the previous surge of machine learning popularity for communication. However, this paper has attempted to cover the significant applications of machine learning for IoT and the relevant techniques, including traffic profiling, IoT device identification, security, and typical IoT applications.

Bibliography

1. Zubarev, Vasily. Machine Learning for Everyone. [Online] https://vas3k.com/blog/machine_learning/.

2. Jain, Rounak. Advantages and Disadvantages of Machine Learning in 2020. [Online] ivy Professional School, Feb 25, 2020. http://ivyproschool.com/blog/2020/02/25/advantages-and-disadvantages-of-machine-learning-in-2020/.

3. *Novel Applications of Machine Learning in Software Testing.* Briand, Lionel C. briand@simula.no : s.n. The Eighth International Conference on Quality Software.

4. Alsheikh, Mohammad Abu. *Machine Learning in Wireless Sensor Networks: Algorithms, Strategies, and Applications.* Singapore : IEEE Communications Surveys & Tutorials, 2014.

5. Ramos, Diana. Real-Life and Business Applications of Neural Networks. [Online] SmartSheets, Oct 17, 2018. https://www.smartsheet.com/neural-network-applications.

6. JavaTpoint. K-Means Clustering Algorithm. [Online] Java T Point. https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning.

7. Aman. K-Nearest Neighbor Algorithm (KNN) in Machine Learning. [Online] analyticsjobs, Jun 02, 2020. https://analyticsjobs.in/education/k-nearest-neighbor-algorithm-knn-in-machine-learning/.

8. Guney, Atakan. Introduction to Bayesian Belief Networks. [Online] towards data science, Nov 20, 2019. https://towardsdatascience.com/introduction-to-bayesian-belief-networks-c012e3f59f1b.

9. Hackerearth. Decision Tree. [Online] https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/.

10. *State-of-the-art deep learning: evolving machine intelligence toward tomorrow's intelligent network traffic control systems.* Fadlullah ZM, Tang F, Mao B, Kato N, Akashi O, Inoue T, Mizutani K. 4, 2017, IEEE Commun Surv Tutor, Vol. 19, pp. 2432–2455.

11. Chandradeep Bhatt, Indrajeet Kumar, V. Vijayakumar, Kamred Udham Singh & Abhishek Kumar. *The state of the art of deep learning models in medical science and their challenges.* s.l. : Multimedia Systems, 2020.

12. Khandelwal, Renu. Deep learning — Deep Boltzmann Machine (DBM). [Online] Data Driven Investor, Dec 16, 2018. https://medium.com/datadriveninvestor/deep-learning-deep-boltzmann-

machine-dbm-e3253bb95d0f.
13. Spirina, Katrine. ow Artificial Neural Networks Can Code Smarter Than GUI Programmer. [Online] Hackernoon, July 12, 2018. https://hackernoon.com/how-artificial-neural-networks-can-code-smarter-than-gui-programmer-1cdfaecb4851.
14. Shuaib, Md Nazmus Saadat and Muhammad. Advancements in Deep Learning Theory and Applications: Perspective in 2020 and beyond. [Online] IntechOpen, March 25, 2020. https://www.intechopen.com/books/advances-and-applications-in-deep-learning/advancements-in-deep-learning-theory-and-applications-perspective-in-2020-and-beyond.
15. missinglink.ai. Deep Learning Long Short-Term Memory (LSTM) Networks: What You Should Remember. [Online] missinglink.ai. [Cited: Feb 19, 2021.] https://missinglink.ai/guides/neural-network-concepts/deep-learning-long-short-term-memory-lstm-networks-remember/.
16. Rajiv. what are the major components of IoT? [Online] RF Page. https://www.rfpage.com/what-are-the-major-components-of-internet-of-things.
17. Little, Kayla. IoT Systems: Sensors and Actuators. [Online] DZone, Aug 6, 2019. https://dzone.com/articles/iot-systems-sensors-and-actuators#:~:text=So%2C%20in%20the%20case%20of,interpreted%20to%20determine%20a%20reading.&text=In%20simple%20terms%2C%20an%20actuator,turns%20it%20into%20physical%20action.
18. *IoT Fundamentals: Networking Protocols, Technologies, and Use Cases for the Internet of Things.* D. Hanes, G. Salgueiro, P. Grossetete, J. Henry and R. Barton, 2017, Cisco Press.
19. *Perception Layer Security in the IoT.* H. A. Khattak, M. Ali Shah, S. Khan, I. Ali and M. Imran. 2019, Future Generation Computer Systems, Vol. 10, pp. 144-164.
20. *Secure and Lightweight Communication in Heterogeneous IoT Environments.* F. Siddiqui, J. Beley, S. Zeadally and G. Braught. 2019, Internet of Things.
21. *RFID technology and its applications in the IoT.* X. Jia, Q. Feng, T. Fan and Q. Lei. Yichang, China: s.n., 2012.
22. A. Koubaa, M. Alves and E. Tovar. *IEEE 802.15.4: A Federating Communication Protocol for Time-Sensitive Wireless Sensor Networks.* 2006.
23. *Survey on Comparison and Research Challenges of IoT Application Protocols for Smart Farming.* D. Glaroudis, A. Iossifides and P. Chatzimisios, 2020, Computer Networks, Vol. 168.
24. *Comparing the Cost-efficiency of CoAP and HTTP in Web of Things Applications.* T. Levä, O. Mazhelis and H. Suomi, 2014, Decision Support Systems, Vol. 63, pp. 23-38.

25. *A comparison of AMQP and MQTT protocols for the Internet of Things.* Nam, Nguyen Quoc Uy and Vu Hoai. 2019, NAFOSTED Conference on Information and Computer Science (NICS).
26. RF Wireless World. AMQP Architecture basics. [Online] https://www.rfwireless-world.com/Terminology/AMQP-architecture.html.
27. AV Systems. A crash course on TR069 (CWMP). [Online] AV Systems. https://www.avsystem.com/crashcourse/tr069/#.
28. *Lightweight Machine to Machine Architecture, Candidate version 1.0.* Open Mobile Alliance. 2013.
29. Wikipedia. Timeline of machine learning. [Online] https://en.wikipedia.org/wiki/Timeline_of_machine_learning.
30. MLK. Brief History of Deep Learning from 1943-2019 [Timeline]. [Online] MLK Blogs, November 24, 2019. https://machinelearningknowledge.ai/brief-history-of-deep-learning/.
31. T, Harwood. Internet of Things (IoT) History. [Online] Postscapes, 2019. https://www.postscapes.com/iot-history.
32. D., VINIK. The Internet of Things: An oral history Web Magazine. [Online] Politico, 2015. https://www.politico.com/agenda/story/2015/06/history-of-internet-of-things-000104/.
33. Rivera, Adonis Tejeda. Machine Learning Basics | What is Machine Learning? | Introduction to Machine Learning. [Online] July 6, 2020. https://medium.com/analytics-vidhya/machine-learning-basics-what-is-machine-learning-introduction-to-machine-learning-f8bbd259f59a.
34. Machine learning. [Online] https://en.wikipedia.org/wiki/Machine_learning.
35. Laizhong Cui1 · Shu Yang1 · Fei Chen1· Zhong Ming1 · Nan Lu1 · Jing Qin2. *A survey on the application of machine learning for the Internet of Things.* Germany: Springer-Verlag GmbH, part of Springer Nature, 2018.
36. Abdulrahman, Adesina. Analytics Vidhya. [Online] Medium, December 7, 2020. Adesina Abdulrahman.
37. Provalis Research - Text Analytics Software Leader. A Brief History of Machine Learning. [Online] June 22, 2017. https://provalisresearch.com/blog/brief-history-machine-learning.
38. Wagh, Sameer Narahari. *" New Directions in Efficient Privacy¬Preserving Machine Learning,* s.l. : A Dissertation Presented to the Faculty of Princeton University, 2020.
39. Tecraze. A treatise on Machine Learning. [Online] TECRAZE. https://tecraze.com/a-treatise-on-machine-learning.
40. LaptrinhX. EVOLUTION OF AI—The HISTORY of MACHINE LEARNING. [Online] June 28, 2020. https://laptrinhx.com/evolution-of-ai-the-history-of-machine-learning-part-2-

2375580721.
41. Foote, Keith D. A Brief History of Machine Learning. [Online] DATAVERSITY, March 26, 2019. https://www.dataversity.net/a-brief-history-of-machine-learning/#.
42. LaptrinhX. Introduction to Machine learning for beginners(PART I). [Online] December 7, 2020. https://laptrinhx.com/introduction-to-machine-learning-for-beginners-part-i-105160678.
43. Aaron Hertzmann, David Fleet and Marcus Brubaker. Machine Learning and Data Mining Lecture Notes. Scarborough: University of Toronto, 2015.
44. Naseer, Muhammad Anser. Machine Learning application in IoT. [Online] Jun 21, 2017. https://towardsdatascience.com/machine-learning-application-in-iot-ff859f9ab4fe.
45. *Using machine learning to refine Category-Partition test specifications.* Lionel C. Briand a, Yvan Labiche b,*, Zaheer Bawar b, Nadia Traldi Spido b. 2009, Information and Software Technology.
46. *Using Machine Learning to support Debugging with Tarantula.* Briand L. C., Labiche Y. and Liu X., 2007, IEEE International Symposium on Software Reliability Engineering.
47. Gralla, Preston. How to incorporate AI and Machine Learning into QA. [Online] Functionize, Jan 15, 2020. https://www.functionize.com/blog/how-to-incorporate-ai-and-machine-learning-into-qa/.
48. StephenMuggleton. *Inductive Logic Programming: Issues, results and the challenge of Learning Language in Logic.* s.l. : ELSEVIER, 1999.
49. Guney, Atakan. Introduction to Bayesian Belief Networks. [Online] towards data science, Nov 20, 2019. https://towardsdatascience.com/introduction-to-bayesian-belief-networks-c012e3f59f1b.
50. Arya, Kuldeep Singh. Decision Tree. [Online] Medium, Mar 4th, 2020. https://kuldeeparya3794.medium.com/decision-tree-73a25d914a2d.
51. *Cybersecurity and the internet of things: vulnerabilities, threats, intruders and attacks.* M. Abomhara. 1, 2015, Journal of Cyber Security and Mobility, Vol. 4, pp. 65-88.
52. *SVELTE: Real-time intrusion detection in the Internet of Things.* S. Raza, L. Wallgren, and T. Voigt,. 8, 2013. Ad hoc networks, Vol. 11, pp. 2661-2674.
53. al, Y. Xin et. *Machine Learning and Deep Learning Methods for Cybersecurity.* s.l. : IEEE Access, 2010.
54. Wang M, Cui Y, Wang X, Xiao S, Jiang J. *Machine learning for networking: Workflow, advances and opportunities.* s.l. : IEEE, 2017.
55. Maslovska, Oksana. Deep Learning: Definition, Benefits, and Challenges. [Online] stfalcon, November 10, 2017. https://stfalcon.com/en/blog/post/deep-learning-what-it-is.

56. Magnimind. Deep Learning and Its Advantages. [Online] Jan 28, 2020. https://becominghuman.ai/deep-learning-and-its-5-advantages-eaeee1f31c86.
57. Li, Xiaochen. *Deep Learning in Software Engineering.* Jinzhou District, Dalian, China: School of Software, Dalian University of Technology.
58. *Easy over hard: a case study on deep learning.* Fu, W., & Menzies, T. 2017, FSE, pp. 49-60.
59. *ActiVis: a visual exploration of industryscale deep neural network models.* Kahng, M., Andrews, P.Y., Kalro, A., & Chau, D.H.P. 24, 2018, IEEE Transactions on Visualization and Computer Graphics, Vol. 1, pp. 88-97.
60. *DeepTest: automated testing of deep-neural-network-driven.* Tian, Y., Pei, K., Jana, S., Ray, B. 2018, ICSE.
61. Hoos, Jesper E. van Engelen & Holger H. A survey on semi-supervised learning. *Machine Learning.* November 15, 2019, pp. 373-440.
62. dshahid380. Convolutional Neural Network. [Online] Medium, Feb 24, 2019. https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529.
63. *Development of PM2.5 prediction model using recurrent neural network.* Park, I., Ho, C. H. and Hur, S. K. 2019, American Geophysical Union.
64. Saulles, M. D. Internet of Things (IoT): Definitions," Information Matters. [Online] February 23, 2017. https://informationmatters.net/internet-of-things-definitions.
65. *Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications.* Al-Fuqaha, M. Mohammadi, M. Guizani, M. Aledhari and M. Ayyash, 2015, IEEE Communications Surveys & Tutorials.
66. DRJ, Arya. IoT - Internet of things Capsule. *ARYADRJ TECHNOLOGIES.* [Online] April 22, 2020. https://www.aryadrj.org/2020/04/iot-internet-of-things-capsule.html.
67. *Internet of Things: Architectures, Protocols, and Applications.* Sarangi, Pallavi Sethi and Smruti R. 2017, Journal of Electrical and Computer Engineering, Vol. 2017.
68. J. Henry and R. Barton. Internet of Things (IoT) Fundamentals. [Online] Cisco Press, 2017. https://learning.oreilly.com/videos/internet-of-things/9780134667577.
69. *Security of the Internet of Things: Perspectives and Challenges.* Q. Jing, A., D. Qiu V. Vasilakos, J. Wan and , J. Lu. 2014, Wireless Networks, Vol. 20, pp. 2481–2501.
70. Performance Lab Software Testing Company. Importance of IoT Testing. [Online] Performance Lab. https://performancelabus.com/iot-testing-importance/.
71. ProfitfromIOT.com . 5 IoT Testing Processes and Open Source Software Tools. [Online] https://iot.electronicsforu.com/ software-dev-tools/5-IoT-testing-processes-tools.

72. Hotify. The top skillsets for an IoT Developer. [Online] https://iotify.io/top-10-iot-skillsets-for-developer.
73. AVsystems. IoT Standards and protocols guide — protocols of the Internet of Things. [Online] AVsystems, May 24, 2019. https://www.avsystem.com/blog/iot-protocols-and-standards/.
74. A. Dunkels, and J.-P. Vasseur. *Interconnecting Smart Objects with IP, The Next Internet.* s.l. : Morgan Kaufmann.
75. F. Hu. *Security and Privacy in Internet of Things (IoT): Models, Algorithms, and Implementations.* s.l. : CRC Press, 2016.
76. J. Henry and R. Barton. "Internet of Things (IoT) Fundamentals, " [Online] Cisco Press, 2017. https://learning.oreilly.com/videos/internet-of-things/9780134667577.
77. *The performance evaluation of IEEE 802.15.4 against IEEE 802.11 with low transmission power.* K. S. Ting, C. K. Ng, G. K. Ee, B. M. Ali and N. K. Noordin. Sabah, Malaysia, s.n., 2011. The 17th Asia Pacific Conference on Communication.
78. S. Tennina, A. Koubâa, R. Daidone, M. Alves, P. Jurčík, R. Severino, M. Tiloca, J.-H. Hauer, N. Pereira, G. Dini, M. Bouroche and E. Tovar,. *IEEE 802.15.4 and ZigBee as Enabling Technologies for Low-Power Wireless Systems with Quality-of-Service ConstTechnologies for Low-Power Wireless Systems Quality-of-Service Constraints.* Berlin, Heidelberg : Springer, 2013.
79. *IoT Communication Protocols: Review.* Shadi Al-Sarawi, Mohammed Anbar, Kamal Alieyan, Mahmood Alzubaidi. 2017. 8th International Conference on Information Technology.
80. *A comparison of 802.11ah and 802.15.4 for IoT.* N. Ahmed, H. Rahman and M. Hussain, 3, 2016, ICT Express, Vol. 2, pp. 100-102.
81. *Securing the Internet of Things: Challenges, threats and solutions,* P. I. R. Grammatikis, P. G. Sarigiannidis and I. D. Moscholios. 2019, Internet of Things, Vol. 5, pp. 41-70.
82. F. Azzola. CoAP Protocol: Step-by-Step Guide. [Online] DZone, Nov 2018, 18. https://dzone.com/articles/coap-protocol-step-by-step-guide.
83. *SecureSense: End-to-end Secure Communication Architecture for the Cloud-connected Internet of Things.* S. Raza, T. Helgason, P. Papadimitratos and T. Voigt. 2017, Future Generation Computer Systems, Vol. 77, pp. 40-51.
84. *Study of Internet-of-Things Messaging Protocols used for Exchanging Data with External Sources.* Ajay Chaudhary, Sateesh K. Peddoju, and Kavitha Kadarla, 2017, IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems.
85. Broadband forum. *Broadband forum Technical Report, "TR-069 CPE WAN Management*

Protocol." Issue 1 Amendment 6. March 2018.
86. *Device Management with OMA Lightweight M2M.* Simon Lemay, Hannes Tschofenig,. Open Mobile Alliance.
87. AV Systems. LWM2M — Lightweight M2M standard — protocol and its benefits. [Online] AV Systems, May 2020. https://www.avsystem.com/blog/lightweight-m2m-lwm2m-overview/.
88. *A survey on the application of machine learning for the Internet of Things.* Qin2, Laizhong Cui1 · Shu Yang1 · Fei Chen1 · Zhong Ming1 · Nan Lu1 · Jing. 2018, International Journal of Machine Learning and Cybernetics, pp. 1-7.
89. Patel HJ, Temple MA, Baldwin RO. *Improving ZigBee device network authentication using ensemble decision tree classifiers with radiofrequency distinct native attribute fingerprinting.* s.l. : IEEE, 2015.
90. *Camera model identification machine learning approach with high order statistics features.* Tuama A, Comby F, Chaumont M. Budapest : s.n., 2016. 24th European signal processing conference.
91. *Do you hear what I hear?: fingerprinting smart devices through embedded acoustic components.* Das A, Borisov N, Caesar M. Scottsdale: s.n., 2014. ACM SIGSAC conference on computer and communications security.
92. *Whom do you sync you are?: smartphone fingerprinting via application behaviour.* Stöber T, Frank M, Schmitt J, Martinovic I. Budapest : s.n., 2013. ACM conference on security and privacy in wireless and mobile networks.
93. *Iot sentinel: automated device-type identification for security enforcement in IoT.* Miettinen M, Marchal S, Hafeez I, Asokan N, Sadeghi A-R,Tarkoma S. Atlanta : s.n., 2017. IEEE 37th international conference on distributed computing systems.
94. *Neural network approach to forecasting the state of the internet of things elements.* Kotenko I, Saenko I, Skorik F, Bushuev S. St. Petersburg, : s.n., 2015. XVIII international conference on soft computing and measurements.
95. *Physical layer authentication of internet of things wireless devices through permutation and dispersion entropy.* Baldini G, Giuliani R, Steri G, Neisse R. Geneva : s.n., 2017. Global internet of things summit.
96. Jincy VJ, Sundararajan S. *Classification mechanism for IoT devices towards creating a security framework.* s.l. : Springer International Publishing, 2015.
97. *A host-based intrusion detection and mitigation framework for smart home IoT using OpenFlow.* Nobakht M, Sivaraman V, Boreli R. Salzburg : s.n., 2016. 11th international

conference on availability, reliability and security.
98. Do VT, Engelstad P, Feng B, van Do T. *Strengthening mobile network security using machine learning.* s.l. : Springer International Publishing, 2016.
99. Xu K, Zhang Z-L, Bhattacharyya S. *Internet traffic behavior profiling for network security monitoring.* s.l. : IEEE, 2008.
100. Hu Y, Chiu D-M, Lui JC. *Profiling and identification of p2p traffic.* s.l. : Computer Networks, 2009.
101. Brauckhoff D, Dimitropoulos X, Wagner A, Salamatian K. *Anomaly extraction in backbone networks using association rules.* s.l. : IEEE, 2012.
102. *Tracked without a trace: linking sessions of users by unsupervised learning of patterns in their DNS traffic.* Kirchler M, Herrmann D, Lindemann J, Kloft M. New York : s.n., 2016. ACM Workshop on Artificial Intelligence and Security.
103. Huang N-F, Jai G-Y, Chao H-C, Tzang Y-J, Chang H-Y. *Application traffic classification at the early stage by characterizing.* s.l. : Inf Sci, 2013.
104. Siryani J, Tanju B, Eveleigh TJ. *A machine learning decision support system improves the internet of things smart meter operations.* s.l. : IEEE Internet Things, 2017.
105. Ling X, Sheng J, Baiocchi O, Liu X, Tolentino ME. *Identifying parking spaces detecting occupancy using vision-based IoT devices.* Geneva: global internet of things summit (GIoTS), 2017.
106. *Early detection of grapes diseases using machine learning and IoT.* Patil SS, Thorat SA. Mysore: second international conference on cognitive computing and information process, 2016.
107. Guo W, Fukatsu T, Ninomiya S. *Automated characterization of flowering dynamics in rice using field-acquired time-series RGB images.* s.l. : Plant Methods, 2015.
108. V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K.Widner, T. Madams, J. Cuadros, *``Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs,".* 2016.
109. *``Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network,".* A. Prasoon, K. Petersen, C. Igel, F. Lauze, E. Dam, and M. Nielsen, 2013. The Medical Image Computing and Computer-Assisted Intervention Society (MICCAI).
110. M. Chen, Y. Zhang, M. Qiu, N. Guizani, and Y. Hao, *SPHA: Smart personal health advisor based on deep analytics.* s.l. : IEEE, 2018.
111. N. Y. Hammerla, S. Halloran, and T. Plötz,. *Deep, convolutional, and recurrent models for human activity recognition using wearables.* 2016.

112. J. Zhu, A. Pande, P. Mohapatra, and J. J. Han, *Using deep learning for energy expenditure estimation with wearable sensors.* s.l. : HealthCom, 2015.
113. A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng,. s.l. : Nature Med, 2019.
114. W.-J. Chang, L.-B. Chen, C.-H. Hsu, C.-P. Lin, and T.-C. Yang, *A deep learning-based intelligent medicine recognition system for chronic patients.* s.l. : IEEE Access, 2019.
115. A. R. Lopez, X. Giro-I-Nieto, J. Burdick, and O. Marques. *Skin lesion classification from dermoscopic images using deep learning techniques.* 2017.
116. *MobileDeepPill: A small-footprint mobile deep learning system for recognizing unconstrained pill images.* X. Zeng, K. Cao, and M. Zhang, `. 2017. 15th Annu. Int. Conf. Mobile Syst., Appl., Services (MobiSys).
117. B. A. Erol, A. Majumdar, J. Lwowski, P. Benavidez, P. Rad, and M. Jamshidi. *Improved deep neural network object tracking system for applications in home robotics.* s.l. : Springer, 2018.
118. S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen. *Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection.* s.l. : J. Robot. Res, 2018.
119. M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh. *Semisupervised deep reinforcement learning in support of IoT and smart city.* s.l. : IEEE Internet Things, 2018.
120. X. Wang, L. Gao, S. Mao, and S. Pandey. 2017.
121. Y. Gu, Y. Chen, J. Liu, and X. Jiang,. ``*Semi-supervised deep extreme learning machine for Wi-Fi-based localization.* s.l. : Neurocomputing, 2015.
122. M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp,P. Goyal, L. D. Jackel, M. Monfort, U. Müller, J. Zhang, X. Zhang,. End to end learning for self-driving cars. [Online] Cornell University, 2016. https://arxiv.org/abs/1604.07316.
123. *End-to-end learning of driving models from large-scale video datasets.* H. Xu, Y. Gao, F. Yu, and T. Darrell. 2017. IEEE (CVPR).
124. B. Wu, F. Iandola, P. H. Jin, and K. Keutzer. s.l. : IEEE, 2017.
125. H. Li, Y. Li, and F. Porikli. *DeepTrack: Learning discriminative feature representations online for robust visual tracking.* s.l. : IEEE Trans. , 2016.
126. *Deep tracking: Seeing beyond seeing using recurrent neural networks.* Posner, P. Ondrú²ka and I. 2016. AAAI.
127. J. Dequaire, P. Ondrú²ka, D. Rao, D.Wang, and I. Posner. *Deep tracking in the wild: End-to-*

end tracking using recurrent neural networks. 2018.
128. D. Singh and C. K. Mohan, `. *Deep Spatio-temporal representation for detection of road accidents using stacked autoencoder.* s.l. : IEEE Tran, 2019.
129. Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y.Wang, *Traffic flow prediction with big data: A deep learning approach.* s.l. : IEEE Trans, 2015.
130. Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu,. *LSTM network: A deep learning approach for short-term traffic forecast.* s.l. : IET Intell, 2017.
131. J. Zhang, Y. Zheng, and D. Qi. *Deep Spatio-temporal residual networks for city-wide crowd flow prediction.* s.l. : AAAI, 2017.
132. J. Wang, J. Zhuang, L. Duan, and W. Cheng. *A multi-scale convolution neural network for featureless fault diagnosis.* s.l. : ISFA, 2016.
133. L.Wang, X. Zhao, J. Pei, and G. Tang. *Transformer fault diagnosis using continuous sparse autoencoder.* s.l. : SpringerPlus, 2016.
134. L. Li, K. Ota, and M. Dong. *Deep learning for smart industry: Efficient manufacture inspection system with fog computing.* s.l. : IEEE, 2018.
135. J.-K. Park, B.-K. Kwon, J.-H. Park, and D.-J. Kang, *Machine learning-based imaging system for surface defect inspection.* 2016.
136. Aaron Hertzmann, David Fleet and Marcus Brubaker. Machine Learning and Data Mining. Scarborough: University of Toronto, 2015.
137. Aurélien, G. *Hands-On Machine Learning With Scikit-Learn & TensorFlow.* . California: O'Reilly Media, Inc., 2017.
138. C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas,. *DDoS in the IoT: Mirai and other botnets.* 2017.
139. *Comparing the Cost-efficiency of CoAP and HTTP in Web of Things Applications.* T. Levä, O. Mazhelis and H. Suomi, 2014, Decision Support Systems, Vol. 63, pp. 23-38.