Interdisciplinary Journal of Information, Knowledge, and Management

An Official Publication of the Informing Science Institute
InformingScience.org

IJIKM.org

Volume 16, 2021

IMPLEMENTING SECURITY IN IOT ECOSYSTEM USING 5G NETWORK SLICING AND PATTERN MATCHED INTRUSION DETECTION SYSTEM: A SIMULATION STUDY

Anshul Jain*	AIIT, Amity University, Noida, India	anshuljain13@gmail.com
Tanya Singh	ASET, Amity University, Noida, India	tsingh2@amity.edu
Satyendra K Sharma	Modern Institute of Technology & Research Center, Alwar, India	skpacific323@gmail.com
Vikas Prajapati	AIIT, Amity University, Noida, India	vikasprajapati1998@gmail.com

* Corresponding author

ABSTRACT

Aim/Purpose	5G and IoT are two path-breaking technologies, and they are like wall and climbers, where IoT as a climber is growing tremendously, taking the support of 5G as a wall. The main challenge that emerges here is to secure the ecosystem created by the collaboration of 5G and IoT, which consists of a network, users, endpoints, devices, and data. Other than underlying and hereditary security issues, they bring many Zero-day vulnerabilities, which always pose a risk. This paper proposes a security solution using network slicing, where each slice serves customers with different problems.
Background	5G and IoT are a combination of technology that will enhance the user experience and add many security issues to existing ones like DDoS, DoS. This paper aims to solve some of these problems by using network slicing and implementing an Intrusion Detection System to identify and isolate the compromised resources.
Methodology	This paper proposes a 5G-IoT architecture using network slicing. Research here is an advancement to our previous implementation, a Python-based software divided into five different modules. This paper's amplification includes induction of security using pattern matching intrusion detection methods and conducting tests in five different scenarios, with 1000 up to 5000 devices in different security modes. This enhancement in security helps differentiate and isolate attacks on IoT endpoints, base stations, and slices.

Accepting Editor Geoffrey Z. Liu | Received: November 5, 2020 | Revised: December 20, December 23, 2020 | Accepted: December 26, 2020.
Cite as: Jain, A., Singh, T., Sharma, S. K., & Prajapati, V (2021). Implementing security in IoT ecosystem using 5G network slicing and pattern matched intrusion detection system: A simulation study. *Interdisciplinary Journal of Information, Knowledge, and Management*, *16,* 1-38. https://doi.org/10.28945/4675

Contribution	Network slicing is a known security technique; we have used it as a platform and developed a solution to host IoT devices with peculiar requirements and enhance their security by identifying intruders. This paper gives a different solution for implementing security while using slicing technology.
Findings	The study entails and simulates how the IoT ecosystem can be variedly deployed on 5G networks using network slicing for different types of IoT devices and users. Simulation done in this research proves that the suggested architecture can be successfully implemented on IoT users with peculiar requirements in a network slicing environment.
Recommendations for Practitioners	Practitioners can implement this solution in any live or production IoT environment to enhance security. This solution helps them get a cost-effective method for deploying IoT devices on a 5G network, which would otherwise have been an expensive technology to implement.
Recommendations for Researchers	Researchers can enhance the simulations by amplifying the different types of IoT devices on varied hardware. They can even perform the simulation on a real network to unearth the actual impact.
Impact on Society	This research provides an affordable and modest solution for securing the IoT ecosystem on a 5G network using network slicing technology, which will eventually benefit society as an end-user. This research can be of great assistance to all those working towards implementing security in IoT ecosystems.
Future Research	All the configuration and slicing resources allocation done in this research was performed manually; it can be automated to improve accuracy and results. Our future direction will include machine learning techniques to make this application and intrusion detection more intelligent and advanced. This simulation can be combined and performed with smart network devices to obtain more varied results. A proof-of-concept system can be implemented on a real 5G network to amplify the concept further.
Keywords	IoT, 5G, network slicing, 5G security, IoT security, IoT ecosystem, intrusion detection system

INTRODUCTION

5G, IoT, Security, Network Slicing, and Intrusion Detection System (IDS) are today's technological buzz words that have consumed most of the space in this insecurely connected world. During the rapid growth of IoT and 5G, we see the latest technological advancements growing along with many issues and vulnerability, one of which is the IoT ecosystem's security. With everyone getting connected to the internet, they are vulnerable to one or other security issues. IoT connected devices are the fastest growing and most exploited in today's advanced technological world.

5G is an upcoming network that will enhance the user experience by providing seamless bandwidth and will undoubtedly be the backbone of IoT. Therefore, we have taken 5G based network slicing as the technological base for this solution, discussed in Jain et al. (2020). Technical specifications of 5G are finalized by 3GPP (2019) and mature day by day with technical advancement and architecture modifications. Some operators have already implemented 5G in the real world. Low power consumption is a requirement for an energy deficit IoT ecosystem, which will get fulfilled by energy-efficient features of 5G. Javaid et al. (2018) discuss the other benefit that 5G will bring: the high density of devices; approximately one million devices can get connected in 0.38 mi^2.

5G also provides network slicing as an essential technological solution. We can use slicing for handling different types of services as discussed in Ni et al. (2018), which is undoubtedly an added benefit for making 5G a more favorable backend network for the IoT ecosystem.

Data generated from transactions happening in IoT will be massive, and securing that data is of tremendous importance. The paradigm of IoT and 5G network has been explained in Wang et al. (2018), where the author explains the successful combination of 5G and IoT, which comprises of "Things", "Internet", and "Intelligence". The author has further advised on the possible use of Machine Learning, Deep Learning, and Artificial Intelligence techniques to improve the IoT ecosystem's performance in 5G.

Today, a significant challenge faced by the IoT industry is a missing standard that lets every other vendor design their solutions; most of these solutions are neither rigorously tested nor robust enough to withstand the hacker's attack. The IoT industry's weakness is a boon for hackers who freely misuse these solutions' prevailing vulnerabilities. Schinianakis (2017) suggests using lightweight cryptographic algorithms as an alternate security solution for 5G and IoT. This traditional solution will help achieve most of the standard security requirements like confidentiality, integrity, and non-repudiation. Using cryptography has its side effects even if it is lightweight; most trusted cryptographic algorithms are CPU intensive, requiring a massive battery for its processing. Jain and Singh (2019) provide a detailed comparison of all cryptographic algorithms used to secure the IoT ecosystem.

The first step to stop an attack is its detection. A security mechanism is said to be effective if it can timely detect any attack or attempt to compromise its resources. Such a solution is technically called an Intrusion Detection System (IDS). Zarpelão et al. (2017) researched several IDS techniques that we can use in IoT. IDS has hugely evolved in the last many years; several high-performance IDS algorithms for pattern matching and information aggregation have been briefly described in Sekar et al. (1999). In this simulation, we are also using pattern-matched IDS to detect the attacks on the network slices.

Here in this paper, we propose a unique security solution using 5G network slicing technology to protect an IoT ecosystem. The solution suggested has used pattern matching or knowledge-based IDS to detect an attack on network slices used for serving end devices.

Many articles mention the benefits of network slicing, such as security, flexibility, and adaptability. Other papers talk about network slicing and its limitations, such as performance and management issues discussed in Ordonez-Lucena et al. (2017), transparency in sharing slicing resources, its brokerage issues in Rost et al. (2017), security issues in network slicing discussed in X. Li et al. (2017), and many others. Barakabitze et al. (2020) raise security concerns using the multi-tenancy approach in slicing networks and highlight slice sharing's potential impact. They further suggest future research to be in this direction to ensure no impact on any slice if any other slice is compromised. This paper addresses these highlighted problems and suggests an inexpensive solution for solving network slicing security problems using pattern-matched IDS.

When a solution gets developed in most scenarios, a new technology comes out, and providing this new technology to everyone could be costly and time-consuming. Contrary to this, when a solution gets developed using technologies already present to everyone, the solution can be advantageous and easy to deploy. Our solution is a python-based simulation suite. It can be easily updated by anyone who wants to make specific changes to the code for their testing and analysis purposes. Setting up an expensive and time consuming 5G network is not required to test our solution's effectiveness.

This research aims to simulate and address part of the security issues for IoT endpoints in a 5G network consisting of base stations and IoT clients using a concept called Network Slicing. The goal is to depict the advantage of using network slicing in a network and its effectiveness in improving network security. Moreover, it is an extension of the research done in Jain et al. (2020), where simulation was conducted manually with limited resources. While increasing the IoT endpoints from 10 to 5000,

several other technological advancements such as pattern matching, intrusion detection, advanced graphical representation have been introduced and implemented in this research. The third section discusses the details of each module, along with flowcharts and diagrams. The simulation results are also discussed and compared to prove that the methodology used is logical and correct.

We have organized this paper as follows; the Literature Review section gives a brief overview of the related work done in implementing security using slicing technology and the gaps identified. This section pens a literature review of all domains of our research and the gaps identified, compared to the research done in this paper. The most critical section, i.e., the third section, explains the detailed methodology used to perform tests. This section also shows how the input data for IoT endpoints, base stations, and network slicing has been collected, generated, verified, and analyzed for genuineness. The third section also explains the implementation of security for IoT endpoints and base stations. This section successfully introduces IDS/IPS to enhance the security mechanism in our simulation. Output generated from the experiments is shown in the form of graphs and logs. The fourth section presents the test results of the five different scenarios performed during the simulation. These five scenarios ranged from increasing test devices from 1000 to 5000 and switching between different security modes. The fifth section compares the results of the test performed in the sixth section. We compared results to prove that the test performed and the results obtained are genuine. Finally, we have presented the conclusion and future work in the last section.

LITERATURE REVIEW AND BACKGROUND

This section discusses some of the most recent studies relevant to our simulation done in this paper. 5G, network slicing, and IoT are research areas that have been in the limelight for quite some time. We highlight a few kinds of research done in IoT security using 5G network slicing. Gaps identified in these studies are detailed below, along with the solutions suggested by the researchers. The later part of this section, along with research questions, discusses and reviews different aspects and history of 5G, security issues specific to the IoT ecosystem, and intrusion detection. It also explains how this paper enhances security, uses network slicing, and integrates it into the IoT ecosystem. This section compares our research with similar research done in this aspect and cites how our research differs.

RELATED WORK AND GAPS IDENTIFIED

Mathew (2020) highlights many network slicing challenges such as Denial of Service attacks, data compromise in multi-tenancy, impersonation attack, and other security issues. One of the several solutions suggested by him includes slice isolation in case of any compromise. The paper further suggests that user equipment that has suffered the impact of slice isolation can be moved to a different slice of the same category and minimize its impact. This suggestion of moving network resources to a different slice has also been implemented in our simulation. 5G Americas (2019), in their white paper, proposes a three-step security algorithm as "Identify isolation mechanism, Establish isolation policy, and Implement policy". We have followed a similar approach in developing and implementing our simulation on the IoT ecosystem. As detailed in upcoming sections and shown in a flow chart of Figure 7, we have combined and advanced most of this research in a different structure where we detect the attack at three different levels, i.e., device, slice, and base station. The methodology discussed in this paper is undoubtedly going to benefit the network and mobile network operator.

Khettab et al. (2018) propose a security architecture that uses both Software-Defined Network and Virtualized Network Functions to enhance security in on-demand slice requirements. The author uses different open-source Intrusion Detection System and Intrusion Prevention System (IDS/IPS) to simulate and test their proposed security algorithms. In this paper, IDS/IPS are signature-based and perform deep packet inspection on the logs generated for the transactions performed. In contrast, in our simulation, we have used Pattern-matched or Knowledge-based IDS/IPS to detect and stop the attack coming on our network. This paper measures the security algorithms' performance with open-source tools, whereas in our research, we detect the actual attack and disables it.

Sattar and Matrawy (2019) propose a solution to handle a Distributed Denial of Service (DDoS) attack on network slices on the core 5G network. The solution suggested here tries to mitigate the DDoS attack by isolating the slice, which otherwise could have adversely impacted the whole network. This experiment further simulates two scenarios, i.e., DDoS flooding attack and slice-initiated attack, to detect and isolate the attacked slice. They propose a solution to optimally allocate and isolate a slice fulfilling all the 5G network requirements. This paper does not mention relocating the users if a DDoS attack goes successful, and the slice needs to be isolated to ensure minimal impact.

Martini et al. (2020) propose a usage control framework for securing the sliced network. One of network slicing's significant and peculiar security problems is the sharing of slices by a similar underlying network. This paper addresses this issue by advanced usage of access control and authorization. The authors have validated their work by implementing a "proof of concept" proposal. Security policy violation gets handled through a very swift response time, and the results showed a minor impact on the user experience. This paper efficiently handles the user access and authentication part. However, it does not mention the scenario if an unwanted user gets access by using social engineering or hacking techniques.

Boutigny et al. (2020) take the research a step ahead. The research targets the 5G network slicing security issue for slices rented to different "tenants" with particular security requirements. This paper addresses two problems in a multi-domain environment where implementation is done within an infrastructure provider and different infrastructure providers where they are unwilling to share the required attributes. It suggests a slice embedded implementation where infrastructure providers cannot share their security-related details and tenants can announce their security requirements. The author also discussed some of the limitations in which they do not support requirements like latency, and the information passed to third-party service providers.

Sathi et al. (2020) suggest securing communication by using security protocols for mutual authentication and one-way authentication. This paper compares "group anonymous mutual authentication" and "one-way authentication protocols" and existing protocols' performance. Mutual authentication protocol helps increase the security related to topology attacks, which can get compromised during slice formation. One-way authentication protocols help increase the security of 5G users during peer-to-peer and device-to-device communication. This paper suggests a method for securing communication but does not talk about identifying any unauthorized access during the whole process.

Several studies suggest user authentication by the mobile network operator as a security mechanism. Saxena et al. (2016) suggest a user authentication mechanism for securing IoT devices on LTE networks with an embedded identity protection mechanism enabled. Kong et al. (2016) suggest using proxy re-encryption for securing communication during a handover session on moving devices. Mahajan and Jindal (2010) explain that proxy re-encryption helps improve session key management generated during handover. Enhancing security using the Public Key Infrastructure (PKI) is a known method of providing secure communication for user authentication, encryption, and hardware devices. PKI-based authentication mechanism has been suggested by Shamir (2000), which requires an escrow account for key generation. Boneh et al. (2001) suggest securing communication using Weil pairing, whereas Gentry (2003) proposes certificate-based encryption, a solution between identity-based encryption and public key infrastructure. Kotulski et al. (2020) explain the security mechanism using slice isolation. They explain different slice isolation types and techniques used in the 5G network. They also suggest the methodology used while practising slice isolation on a Radio Access Network and Core Network. They entail different properties, parameters that can be used in calculation while deciding the network or node level's isolation levels.

Proper authentication of any user or service is an integral part of ensuring security in any domain, and the same applies to the 5G network and IoT ecosystem. Ni et al. (2018) propose a framework for service-oriented authentication, where fog nodes can choose a secure network slice for communication. In this solution, users can anonymously authenticate to ensure the security and confidentiality

of data. The paper also suggests a three-party authentication mechanism using secured sessions key generated using the Diffie-Hellman algorithm. As a future direction, they suggest a privacy-preserving authentication mechanism for moving user's equipment. Our simulation has taken care of this direction in this research. Any possible impact on moving devices is detected, and either device is disabled, or if the slice is isolated, the device gets located to another similar slice.

A related study in a different direction was done in Kotulski et al. (2018), explaining a graph-based slice isolation model. Our study's architecture has five layers, and each layer has a different aspect of network slicing and isolation, which can handle end-to-end slice isolation. The highest layer here represents the leading network, i.e., Radio Access Network and Core Network, whereas the lowest layer represents the network's actual and virtual resources. The algorithm suggested here calculates the best method for slice isolation if any vulnerability is detected and calculated using the graph vector method as a future research direction. The author has suggested developing a fixed set of properties for devices and links considered in our simulation.

Sciancalepore et al. (2018) suggest a new solution for slice allocation. They call it "ONETS: An Online NETwork Slicing solution," wherein they proposed an algorithm that will act as a slice broker. This broker will combine its previous data to gain past slicing knowledge and implement it for future allocation, maximizing resource usage. They further implemented the solution on commercial hardware as a proof-of-concept to validate the feasibility of the solution. This solution has successfully experimented on slices. If an attack or compromise happens on the base station, this part still needs to be handled separately. This gap is taken care of in the simulation done by us.

In Boussard et al. (2018), the authors propose a secured home network based on a software-defined virtual network. Security there gets achieved by isolated and usage initiated slicing technology. They define the future network, which is well advanced, but the authors warn about the underlying security concerns of the Software Defined Network (SDN), which must be handled carefully. Our research fulfils this aspect of the SDN by providing a solution that we can use in different scenarios as per the user's requirement. Two main facilitators for network slicing in 5G are Multi-access Edge Computing (MEC) and network slicing. However, as mentioned earlier, they have many known security issues. One most common solution to this is slice isolation, as suggested by Ksentini and Frangoudis (2020). It suggests a solution for handling traffic redirection to ensure that unauthorized users do not access the network during traffic redirection.

Blockchain is the technology that ensures data integrity and security. Togou et al. (2020) have suggested a signalling-based network slicing framework, which they call a distributed blockchain. Here, they ensure that all the blockchain's technological advancements such as storing and matching hash for all the incoming requests so that any unauthorized user does not get entry to the network. It helps improve the user's experience and parallelly safeguards that security aspect. Blockchain use cases are discussed in Jain et al. (2021). These use cases and the induction of security protocols can get implemented on the 5G network. However, implementing blockchain on a vast 5G network could have a considerable cost impact on the resource.

Usage of machine learning techniques is becoming quite typical for detecting attacks on any network. The same applies to 5G network slicing. Authors Liu et al. (2020) discuss some of the learning-assisted solutions for slicing and evaluating their performance under Denial of Service (DoS) attacks. Network slicing security is analyzed on a single and multiple network node, after which its performance gets evaluated and suggested solutions to mitigate the DoS attack. Further, a system prototype gets developed and experimented with to test the solution. The scenarios covered in this research do not mention what needs to be done to minimize users' impact and the network if any attack goes successful.

5G and Its Security Issues

The mobile network has traveled miles in terms of speed and technology, as shown in Figure 1 since its inception from First Generation (1G) technology in 1980 to Fifth Generation (5G) in 2020. Pirinen (2014) mentions that the 1G network's data speed was approximately two kbps, growing up to 500 Kbps in 2G, 30 Mbps in 3G, and up to 1 Gbps in 4G. Researchers have claimed that in 5G we would achieve a speed of up to 20 Gbps, which will make all the far-sighted virtual technologies happen in real-time. 5G network is a fully virtualized cloud-based network that will disrupt the market hugely and positively. With the successful deployment of 5G, we will also see the real-time deployment and use of the buzz word "Smart" used with every other word like cities and health. With its capacity to handle almost 100 times the number of devices from its current predecessor with significantly less or no latency and high data transfer rates, 5G will help boost the deployment and widespread use of real-time services such as gaming, health, HD-video, IoT devices, and others. This statement gets supported by a study done by GSMA (2017b), which estimates that there will be "1.2 billion 5G connections by 2025, covering 40% of the global population or about 2.7 billion people".

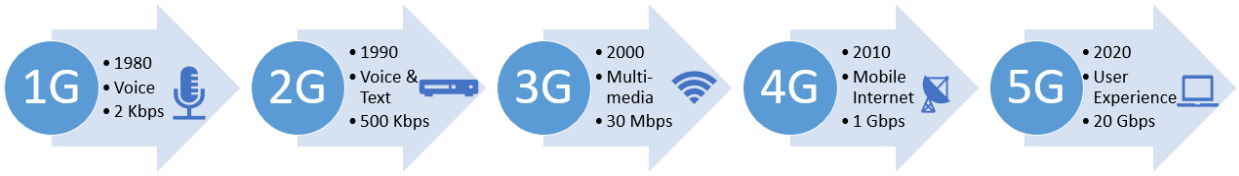

Figure 1. Growth of wireless technology from 1G to 5G

The fifth-generation (5G) of mobile technology is the latest and the fastest technology available for mobile users' speed. We consulted Tutorials Point (2020), which says that 5G speed can go up to 20Gbps. It is a disruptive technology that is going to change the whole paradigm of the current mobile network. 5th generation wireless technology is an amalgamation of all upcoming technologies and innovations. The future ahead lies in the Internet of Things. 5G will enable every machine to talk to the other machine through a less than 1mm frequency.

Moreover, this will exhibit very low power usage, high battery life, and the potential to get charged with sustainable energy availability such as solar energy, wind energy, or else energy from the machine running around. The world is progressing towards a culture where things would be sensing, actuating, communicating, and controlling other things using the internet, and the challenges are manifold. There was a time when every action needed human control, but several machines later started working smartly. An intelligent system will be required in the next few years to operate both with and without human participation. 5G is not an evolution of 4G. In a certain way, if we see, 5G would enhance the features of 4G in terms of high-speed communication and low power consumption. This intelligent system should comprise intelligent internetwork, intelligent data flow, analytics, and security of the data flowing through the network and between the machines while mutual exchange. Various applications and use cases for 5G are:

- **Enhanced Mobile Broadband**
- **Massive Machine Type Communications**
- **Ultra-Reliable/Low Latency Communications**

Brahmi (2013) mentions 5G as a fully converged, heterogeneous, and interoperable software-based wireless network designed to enhance the user experience by inducing the most complicated features, increasing bandwidth, and providing backward compatibility with its predecessors 2G, 3G, and 4G. The evolution of 5G, as displayed in Figure 1, shows its convergence with underlying 2G, 3G, and 4G technology to new wireless technology as explained in Fang et al. (2014). 5G's evolution depicts diverse nature while being compatible with its underlying technologies.

5G will support IoT devices on a large scale. Therefore, designed to combine and support 6 GHz and the frequency band beyond 6 GHz, enabling its support from UHD videos to IoT devices with low data requirements as suggested by Oshin et al. (2016).

Internet of Things (IoT) has been growing exponentially since its inception a decade ago. Benefits of 5G, as discussed in Tutorials Point (2020), and shown in Figure 2, will further help boost the growth of the IoT device market.

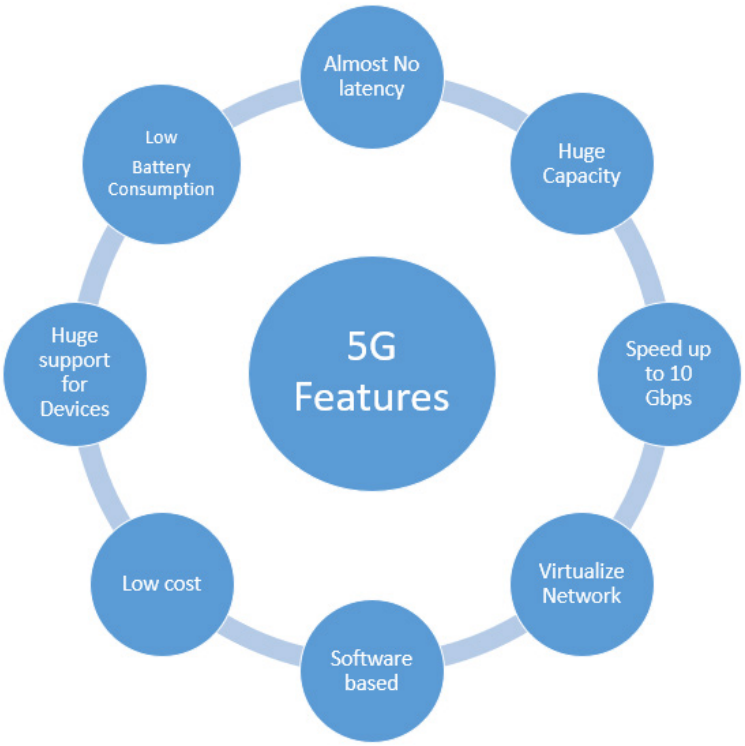

Figure 2. Benefits of 5G mobile technology

5G is a cloud-based virtualized network, a software-based solution, where all its hardware and software resources are application-based. As defined in Afolabi et al. (2018), there are two main parts,

> **Network Softwarization**: This means that all the components involved in the network, from deploying to managing, designing to operations, are all software-based. Their properties can be changed on the fly using software solutions suggested by Nakao et al. (2017). Network Softwarization helps in wide adaptability and elasticity in network reconfiguration.
>
> **Network Slicing:** Provides logically isolated network resources that allow cloud like multi-tenant solutions that get deployed over an entire network. Isolation features further enhances the security of the network.

Network slicing is the base of the simulation done in this paper, and slicing is further discussed in the next section to look at it in more detail.

Handling 5G network issues is much more critical because it is a software-based virtualized network, and, therefore, it inherits all the existing security issues within it. Other than software, Mantas et al. (2015) mention user equipment, core network, and external network as the most vulnerable target of any 5G network. Details of these are mentioned below:

External Network is very vulnerable to attacks such as Denial of Service, or Distributed Denial of Service, as F. Li et al. (2013) mentioned. Attacks can be made successful through 5G handsets by directly accessing the operator's resources or installing malware on the device and causing disruptions on the live network. As 5G is an IP based network, it is more vulnerable to such attacks.

Core Network is the backbone of any wireless telecom network. Any minor attack on this might cause a significant impact on the whole network. 5G network is a heterogeneous network which will provide backward compatibility to 2G, 3G, and 4G network. Jover (2013) says it will also inherit all the primary and prevalent issues in older technologies. Providing backward compatibility will make this network more complicated, hence, making room for security issues. Some of the known attacks or vulnerabilities in 4G are (Forsberg et al., 2007; Seddigh et al., 2010) user equipment location tracking and bandwidth theft by modifying the behaviour of supporting algorithms, message insertion. One typical attack applicable to all is a physical attack on base stations (BTS/enodeB, gnodeB). DoS/DDoS, which gets mentioned as an attack for an external network, is also applicable to the core network.

Last and the most common attacking method on any network is using its customers and their user equipment (UE). As discussed in the core network, 5G will be supporting heterogeneous technologies. Therefore, UE's will also get the operator's underlying support, which will enable attackers to make them the soft target. Innocent users get lured to the open-source software and download free software for games or other attractive applications with inbuilt malware. Such malware can target the operator and its network, base stations, or even go to the extent of causing DoS attack, as discussed in Becher et al. (2011).

Network slicing is discussed in the next section. One of the solutions designed in the 5G network minimizes such attacks by isolating that specific section of a slice. Our simulation will provide a solution to such types of attacks where either device, slice, or a base station gets compromised. We will detect the impacted section and disable it, whether it be an IoT device, a base station, or a full slice. Our solution will provide us with a quick and less expensive solution to isolate the impacted or vulnerable section of any network.

Security Issues in IoT Ecosystem

We have already discussed security challenges related to 5G in the section above, where we saw that challenges exist everywhere from the core network to the internal and external network. IoT ecosystem is also a part of this big picture. Many challenges co-exist with prevailing security issues on any network. In this section, we will take a brief overview of the essential security issues concerning IoT. As shown in Figure 3, the IoT ecosystem gets divided into three layers, i.e., the perception or physical layer, the network layer, and the topmost layer as the application layer (Oshin et al., 2016; Schinianakis, 2017).

The perception layer is the bottom layer, which is sometimes called the physical layer. The name suggests it directly deals with all the physical components in IoT infrastructure, and therefore, issues related are similar. This layer deals with physical access to devices like device theft and tampering, data sniffing through direct wire access, terminal security, RFID hacking, and many others. Forsberg et al. (2007) discussed the solutions to these problems. They provide solutions such as using chip-level security by implementing a Trusted Platform Module (TPM), tamper proof shielding on outdoor devices, PKI, and data encryption.

The network layer is the middle layer, and it deals with all the network-related communication technologies like Wi-Fi, 2G, 3G, 4G, 5G, Bluetooth, and protocols like IPv6. Virtual Private Network and Next Generation Networks get integrated at this layer. Some of the security issues reported in this layer are routing issues, routing protocols issues, internet security, bandwidth, LAN security,

compatibility issues, data on travel issues, and many others. Forsberg et al. (2007) suggest security solutions using robust routing protocols, IDS/IPS, firewall, VPN. Jain and Singh (2019) discuss data encryption and restricted access control on network devices.

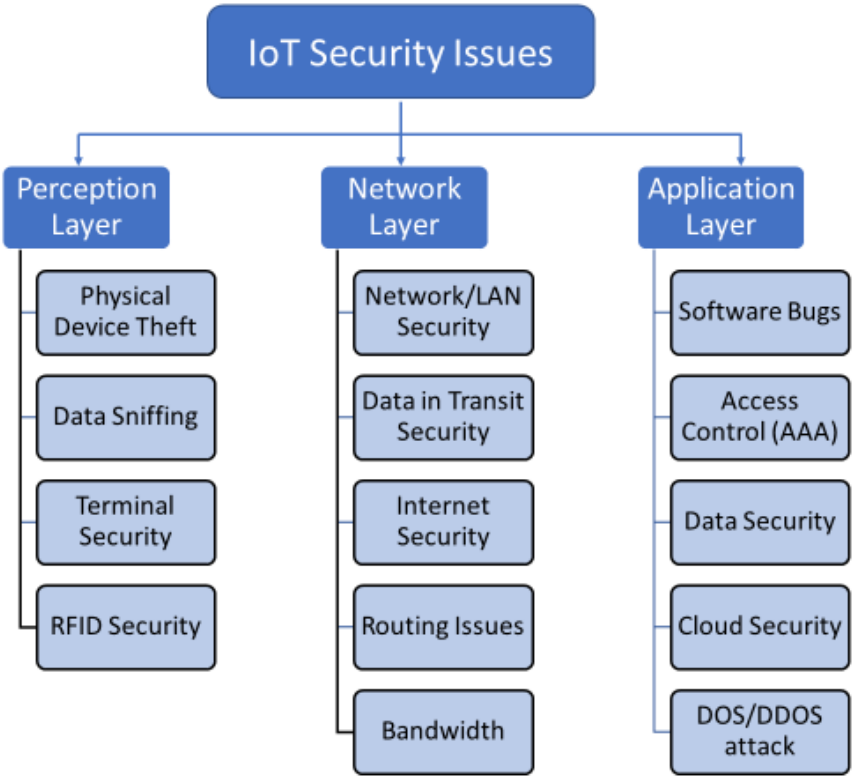

Figure 3. Security issues in IoT layers

The third and topmost layer in the IoT ecosystem is the application layer, and this layer directly interacts with the end-user; therefore, it is most vulnerable to attacks. All the latest technologies like artificial intelligence, data mining, cloud computing, and IoT applications managing their devices lies in this layer. As independent users develop the applications for managing IoT devices without following any security standards, they are most vulnerable, leading to network compromise. Jain and Singh (2020) discuss security issues such as software bugs, data security, cloud security, authentication control, web service security, secure data mining, middleware security, and traditional attacks like DoS/DDoS, eavesdropping. Few solutions suggested for this layer are using IDS/IPS, certifications, user authentication and authorization, regressive application testing, and trust management.

In this simulation, we would be providing a security solution through an application that will target the middle layer using network slicing. Network slicing is a virtualization technology that is widely adopted and implemented in 5G. This experiment will also provide a solution like IDS/IPS by detecting the attacks and disabling the impacted devices to restrict their impact to a smaller network section.

Intrusion Detection in IoT Ecosystem

Al-Bahri et al. (2018) write that, with almost 75 billion connected IoT devices, security will always be a concern. The first step to isolate and remove any security issue is to detect it. IDS is a methodology used to detect security issues. The next step is to prevent the network's attack by using intrusion prevention techniques. Threats are of two types, new or unknown threats and the known threats like DoS/DDoS. IDS/IPS systems are the best solution for detecting both types of attacks, as

Chaabouni et al. (2020) discussed. For new attacks or unknown threats, we might require manual or human intervention. Nowadays, we can see a new integrated system consisting of IDS and IPS, called IDPs. IDPs is an intelligent system that can detect and prevent incoming intrusions on the network. Chaabouni et al. (2019) discuss that new systems are enabled with intelligent machine learning techniques, which develop their understanding and intelligence based on the ongoing learnings.

IDS get classified into Host-based IDS (HIDS) and Network-based IDS (NIDS). An IDS whose monitoring activities are limited to the transactions performed on any single host or computer is called HIDS. Krishna and Tyagi (2020) write that these systems are generally installed over all the computers in an organization to detect malicious activities. A NIDS system is generally installed on the network and can monitor a large part of it. This system monitors the data at the packet level; however, such systems cannot monitor encrypted data. NIDS gets installed on top of existing network equipment like routers, switches, or firewalls.

Eskandari et al. (2020) mentioned two categories of IDS that generally detect malicious traffic based on two methodologies, i.e., Behavior-based and Knowledge-based. According to Scarfone and Mell (2007), behaviour-based systems are intelligent and statistical systems that build their baseline over a period, based on the traffic going through it, based on the developed baseline system that can differentiate between normal and abnormal traffic. Such systems intelligence increases as more and more traffic pass by, and they can build their database. The second and most common IDS methodology is a Knowledge-based system; it gets referred to as Signature or Pattern-matching Detection. This system has a ready-made set of already known datasets, against which incoming and outgoing traffic gets matched, and alerts get raised. Such types of IDS cannot detect new or unknown vulnerabilities, also called Zero-day vulnerabilities.

In this simulation, we would be using a Knowledge-based IDS, also called Pattern-matched IDS. This methodology has defined a few fixed parameters against which all the incoming traffic gets matched. In case any deviation gets noticed, an alert gets raised, and the impacted resource is disabled. Figure 7 shows the detailed process flow chart of this implementation.

Network Slicing and Its Integration in IoT

Andrews et al. (2014) discuss that the 5G network is a fully virtualized cloud-based network that is easily expanded and modified as per the real-time user requirement. Nanda and Tzi-cker (2005) write that network slicing is based on virtualization, which has its root back to 1960, whereas Goldberg (1974) says IBM introduced slicing in an operating system. Since then, virtualization has emerged as a market where most organizations worldwide are now using it or are planning to migrate their systems to cloud-based computing.

According to Next Generation Mobile Network (NGMN) (2020), slicing is an old concept that has been defined in a new way in 5G network technology. Hedman (2016) categorized network slicing into three layers: service, network, and resource layers; each of the three layers has its specific functionalities. The service layer is the topmost layer that directly interacts and serves the application or mobile network service provider. The network layer is the middle layer responsible for most network-related tasks and can enhance performance. This layer can also contain a sub-network layer, which gets further shared with other services or functions. The bottom-most layer is the resource layer directly interacting with other resources, network infrastructures, or functions.

Some of the main principles of network slicing, as mentioned by Afolabi et al. (2018), are Automation, Isolation, Customization, Flexibility, Programmability, Hierarchical, and End to End. Out of these seven principles, network slicing works on isolation. We will discuss isolation as this the principle that enables and ensures security in the network. Nakao et al. (2017) say this principle provides a security guarantee to every tenant that the slice they are using is safe from other network conflicts. Implementing slicing in a 5G network is a resource-consuming task, and it adversely impacts multiplexing gain. Like cloud-based solutions, slicing can be implemented in multiple ways in a network,

like a deployment on different hardware resources, different shared resources, or a shared resource-restricted with administrative access policies.

As shown in Figure 2, there are many features of 5G which vary as per requirement. However, one of the most critical services that 5G will cater to is the unmatched user experience, including low bandwidth requirement for IoT devices, low power consumption, virtual reality, and deliver content in Ultra High Definition Video. Fulfilling them all together is a mammoth task that is possible using network slicing.

Slices are the virtual distributions of the available resources on a network, ensuring that the usage gets maximized and the impact gets minimized. Figure 4 shows a fundamental distribution of a slice in any base station. Slices can be allotted based on the user's requirements such as each slice can hold a unique tenant, a base station, service, or any device.

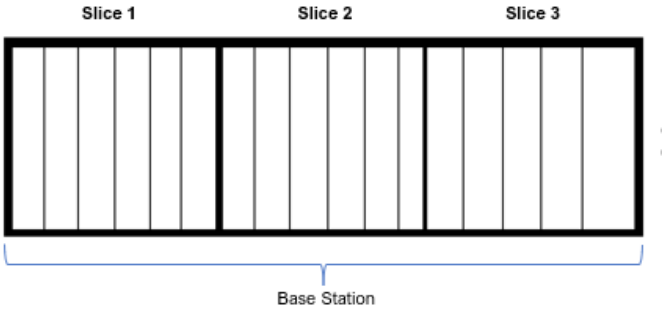

Figure 4. Base station sliced representation.

We have a unique benefit from using network slicing, especially for IoT devices, because it requires low power wireless technology to support the devices in a wide area. As slicing is a cloud-based solution, Modi et al. (2013) say that clouds' underlying and known security issues also migrate into slicing technology. Therefore, slicing technology, implemented as a security solution in 5G, also brings security issues addressed in this research. Improvement in technology makes it more complex, and the security challenges in network slicing are also complicated and multi-faced. Afolabi et al. (2018) discuss that one of the prevalent challenges is the compatibility of different network resources and incorporation of segments within the same area. Other security issues come from the cloud's technology inheritance, like sharing resources among different tenants and their exposure. The higher the exposure a tenant has, the higher the risk for others. A similar study, Hedman (2016) on Next-Generation Mobile Networks, also points out that sharing slice resources between different tenants is a critical and vital security issue. Among other challenges related to using slicing are explained by X. Li et al. (2017). They also highlight that dynamic creation, allocation, and management of slicing and isolation of resources within slices is also a significant security concern. In this software simulation, we are targeting to solve such underlying security issues. We have divided resources into three parts "slices", "IoT devices or clients", and "base station". We will be detecting either the impacted resources, slices, or base station, and those detected resources will be immediately disabled to avoid escalation of impact.

PROPOSED ARCHITECTURE AND DESIGN METHODOLOGY OF IOT ECOSYSTEM IN 5G WITH NETWORK SLICING

PROPOSED ARCHITECTURE

5G is a cloud and software-based network; therefore, all the underlying security issues of a standard network and its software get inherited in it, and the same is discussed in Singh, Verma, S., and Parashar (2016). These security issues can exploit its resources in different ways, many of which can be Zero-day vulnerabilities. Detecting intrusions on cloud networks is possible, as discussed in Singh, Verma, Kulshrestha, and Katiyar (2016). 5G design will match the expected user experience required for today's latest gaming solutions, IoT, and VR solutions. Therefore, handling security issues becomes parallelly essential to provide users with an uninterrupted experience.

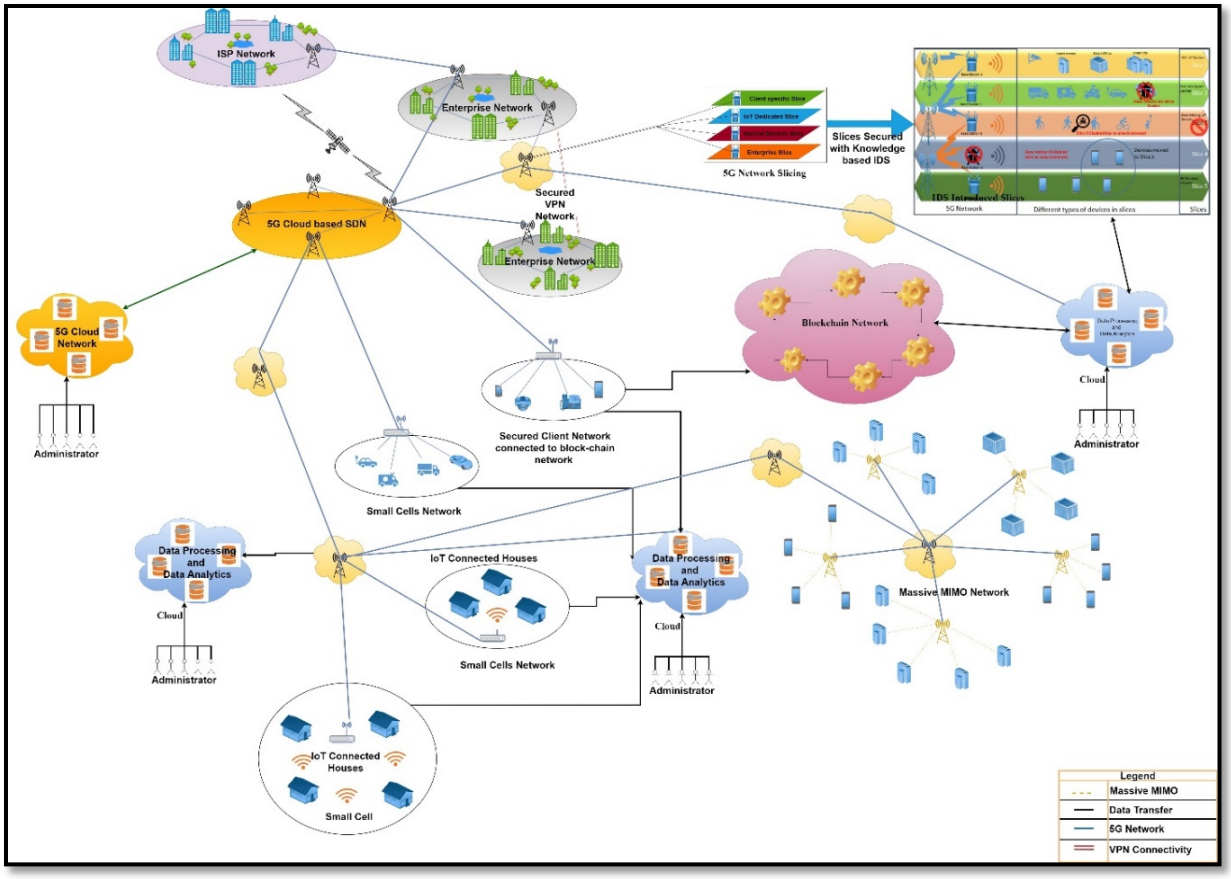

Figure 5. Proposed 5G-IoT ecosystem architecture

We proposed the IoT ecosystem's architecture in a 5G network using the slicing module as designed by us. All the other underlying network architecture would remain the same and will benefit from it. In this simulation, we have proposed a security solution using network slicing, and the same is visible in the architecture proposed below. We have connected the network slice to a 5G network on which this simulation is performed. As discussed in the previous section, network slicing is one of the technologies that we would be used to detect and isolate the security issues in this simulation. Zhang et al. (2016) say that 5G is heterogeneous, and Sun et al. (2015) say that 5G is a software-defined network supporting the underlying networks and the other latest technologies. Therefore, as shown in Figure 5, the proposed architecture consists of ISP network, Enterprise network, 5G network, 5G

cloud, Blockchain network, Small cell network, Massive MIMO network, and 5G network slicing. The proposed network slicing methodology, its security application, and its implementations are discussed in this paper's following sections.

Proposed Design Methodology

This simulation's design has been developed in Python, which comprises five different modules, as shown in Figure 6. Modules are named as "Data Input", "Main Module", "Security Module", "Intrusion Detection" and "Output." Each component is an integral part of the simulation test and has a significant dependency on each other. The design methodology proposed in this paper is an extension and advanced research of simulation exhibited in Jain et al. (2020), where the authors performed a manual simulation with ten devices. This methodology has included many new modules such as intrusion detection, advanced output module, advanced slicing module, and increasing the hardware capacity to test up to 5000 IoT clients.

Figure 6. Design flow and modules

The essence of the methodology used in this simulation lies in implementing security, which will be done by configuring, enabling, and testing security and attack mode.

Input-file: This comprises the YAML format data provided to the main module on which the program would generate output.

Main Module: The main module consists of the programming part that would read the input data, perform processing, and give the output.

Security Implementation Module: This module enables the security feature in this simulation module. The module here is a configurable module that can be enabled or disabled as per the user's requirement. This module gets further divided into two parts with subparts.

Attack/Intrusion Detection Module: This module consists of detecting attacks on the security implementation module's three resources. These attacks will be identified based on 16 other configuration parameters. Mismatch in the values of these with real transactions will cause an alert to disable the resources.

Output: The output consists of results in graphical and text format that this simulation suite will generate.

Figure 7 explains the whole process of implementing this simulation using a flow chart. We divided the entire process of simulations into three segments:

i) The first segment will collect and validate data for its accuracy, and the same data will get forwarded to the simulator to verify if it is working as expected. The output gets generated in the form of graphs and logs.

ii) The second segment, the most essential and critical module, performs most of the experimentation task. Here security and attack mode will be turned on to perform simulation in different scenarios. The impact will be detected and analyzed based on 16 different predefined patterns and different resources like IoT devices, base stations, and slices.

iii) The third and last segment will replay the security module to detect if:
   a. The network still has any compromised resources.
   b. All resources are functioning correctly, after which it will generate the reports.

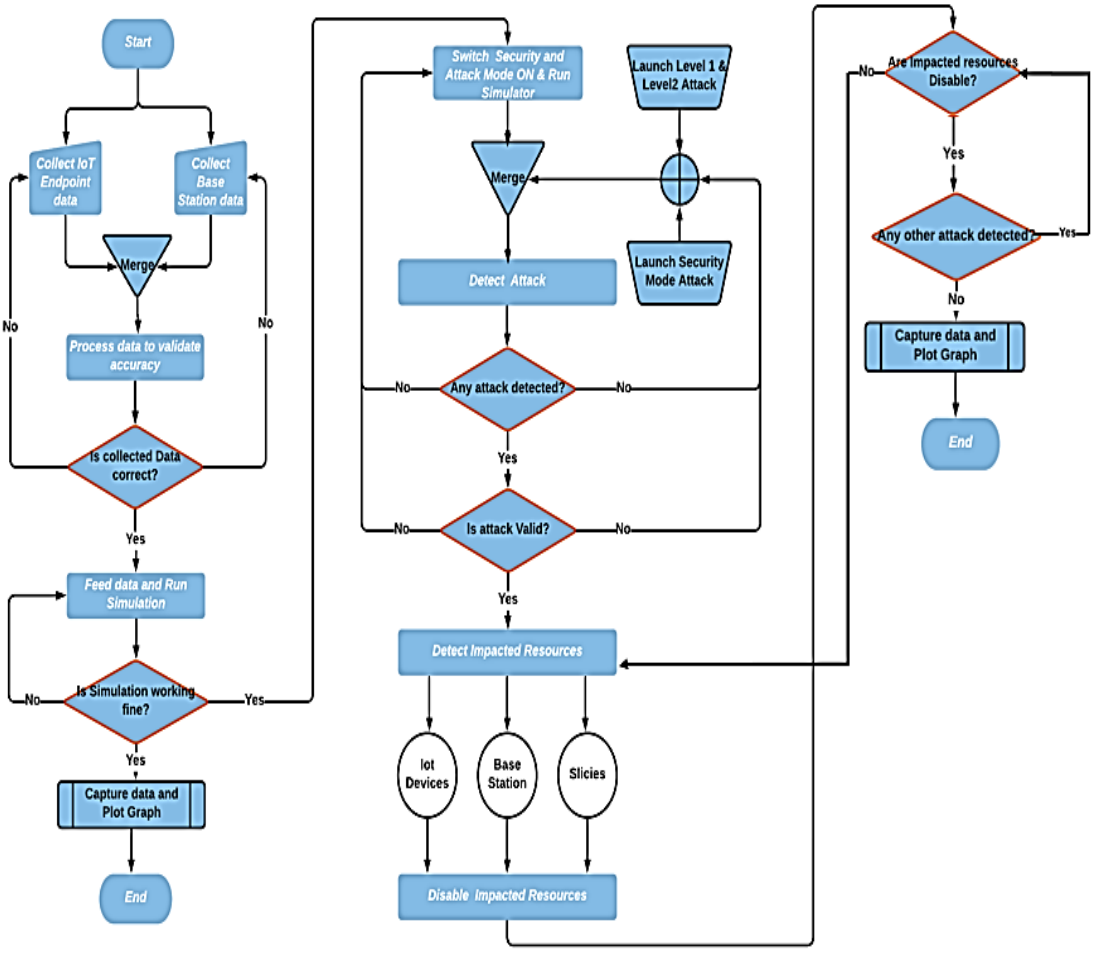

Figure 7. Detailed flow chart of the simulation process

The following sections will discuss each of these modules individually with the detailed technical solutions used.

Input File Design

The input file of the test is designed to write in YAML format. YAML uses indentation to define structured data where the data blocks get differentiated by white spaces only.

The below section discusses an input file that contains data for base stations and different IoT endpoints.

IoT endpoints

Two crucial factors for deciding IoT endpoint behaviour is mobility and usage pattern. The flowchart in Figure 8 shows the process used to collect the mobility and usage pattern data used in this simulation.

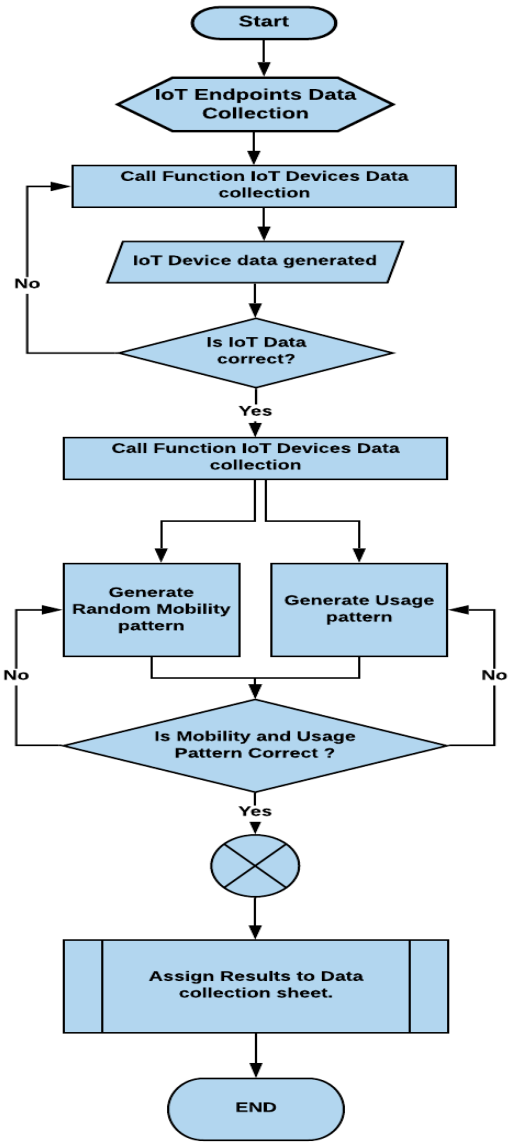

Figure 8. Flowchart for making a list of "Usage and Mobility Patterns"

Mobility Pattern: Mobility pattern specifies at what rate the IoT devices would move. For example, if an IoT device is stationary, then the movement pattern will be 0 because there is no movement. In contrast, for the devices attached to a car, tram, or the ones walking, the mobility pattern must be calculated.

For mobility patterns, the first thing to identify is IoT endpoints (users) for which the mobility pattern is required. IoT endpoints identified include moving devices like cars, devices attached to walking people, stationary people. Subsequently, their parametric fields are decided, including their distribution pattern, minimum and maximum value, and the total population ratio.

Distribution: This parameter specifies how the clients get generally distributed in a particular area. While the people (walking, stationary, and slack) got randomly distributed, the tram and cars got evenly distributed.
Parameters: This specifies the minimum and the maximum value
Client Weight: It is the ratio of the type of client in a group of 100 people.

Usage Pattern: This factor signifies the rate of usage. For example, if an IoT endpoint requests for the voice facility from the base station, then at what rate bandwidth gets consumed and at what time.

Following is the process to be followed for every IoT endpoints:

Get IoT device "loc_x", "loc_y" from the client block of the YAML file.
Calculate the location of client (using get_dist() in build function).
Randomly get the mobility pattern of the client
Randomly get the usage pattern.

All these values are collected and appended to the client module to create an instance of the client. This instance serves the client's purpose in the real world, and it has a "moblity_pattern" module for moving resources, "usage_pattern" for checking consumed resources.

Base station

After calculating the mobility pattern and usage pattern, the next step is to dispense the bandwidth to the base station slices. In this step, the base station gets iterated, bandwidth gets calculated for each slice, and a container gets initialized. When the user gets connected to the base station, the IoT device will request a service, and the base station would have to allocate the resources for that slice. So, instead of allocating resources, a container is apportioned that contains the resources. This container makes sure that while the device gets connected to the base station, the assigned resources are not shared out to another device. It contains the resources and the IoT device for the time till the device does not get disconnected.

The calculated information gets appended to form a list of "Slices". Once slice bandwidth is calculated, this information plus other information is passed over to the base station module to create an instance of a base station initialized to variable "base_station". This "base_station" depicts the base station in the real world.

Below is the description of the parameters identified for the slice block.

Type of slice: Use cases as mentioned in X. Li et al. (2017) are (eMBB (enhanced Mobile Broadband), URLLC (Ultra-Reliable Low Latency Communications), and mMTC (massive Machine Type Communications)) and voice.
Guaranteed bandwidth: Guaranteed bandwidth defined the minimum assured bandwidth that each user would connect to the services.
Maximum bandwidth: This defines the maximum bandwidth that the user can get when the user's consumption increases.
Quality of Service (QoS): It is a value between 1-7 used to prioritize the service. A higher value (5,6,7) means less significant traffic, and less value (1,2,3) specifies that the traffic is of higher priority.
"x_val" and "y_val" are required. These values are the minimum and maximum values of the x and y of the considered area.
The limit for the number of base stations is defined.

Deciding parameters for the base station are:

Location: These are the (x, y) coordinates of the base station.
Coverage area: It is the coverage area of the base station in meters.
Throughput: Amount of data transmitted by the base station.

Slices: Consist of the slices for which the base station provides services (discussed in the previous section).

Simulation (Main Module)

The main module is a system of a discrete sequence of events that is an asynchronous part that reads the data from input-file, i.e., YAML file, and processes the data to generate the output (graph and the log file). The primary entity of this testing is the IoT endpoints and base stations. For ease of management, modules were decoupled and divided into smaller sub-modules. The design of the main module is as shared in Figure 9.

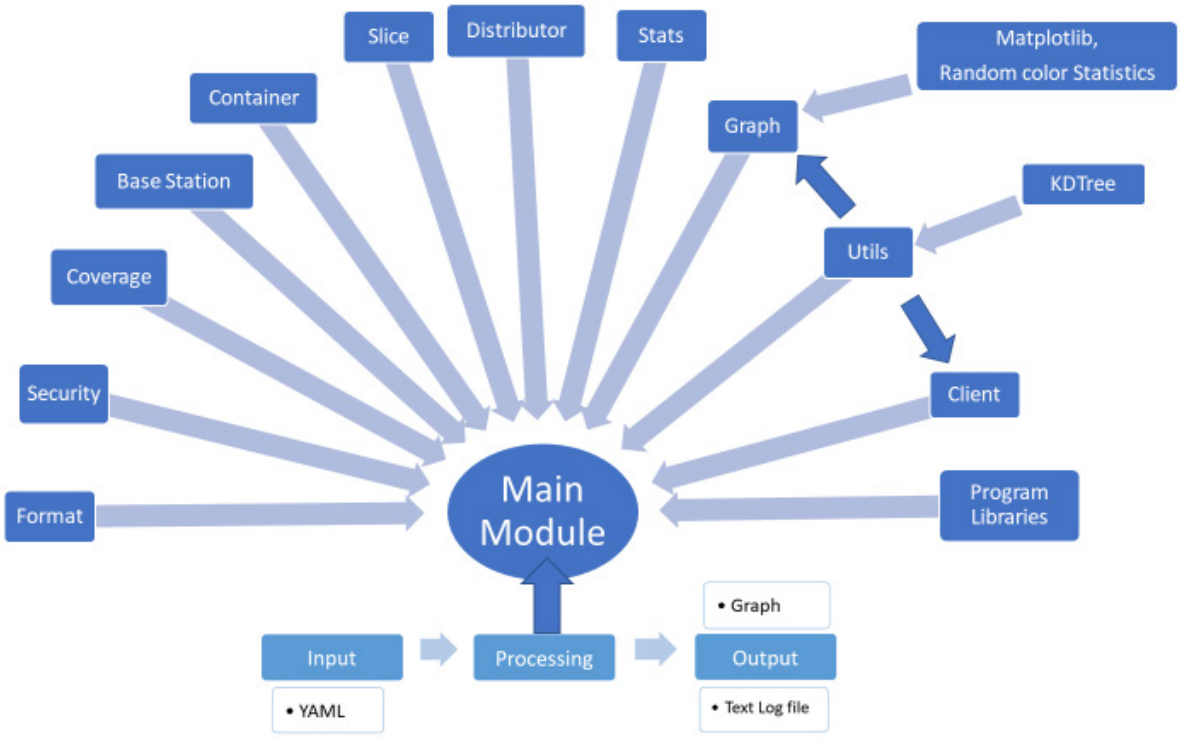

Figure 9. Main module design format

Figure 9 shows inbuilt libraries and modules that are combined using the main module. Each module is discussed below in brief with its functions.

Main Module: This module is the heart and incorporates all the modules developed to streamline the process. This module consists of functions such as data calculation, instances creation for base station and clients, and simulating to generate statistics.

Format Module: This function converts bits representation to bytes, kilobytes, megabytes.

```
def format_bps(size, pos=None, return_float=False):
```

Security Module: This function is one of the most important modules of this simulation. It implements all the security functions discussed in the 'Security Implementation' section of this paper.

Coverage Module: This module is an extension to the base station module, but we have developed it as an individual module. This class describes the two essential properties of the base station,

i.e., centre and radius. Both values are in the form of coordinates (x, y). The centre's value gets randomly generated, whereas the value of the radius is provided manually. This value helps to initialize the data member of the coverage class.

Base Station Module: This class is a specimen of the real-world base station. It is sub-divided into two other modules, namely, coverage and slice.

Container Module: The container module is used for containment purposes. Here, the base station defined has slices(services) requested by the client, allocated as per the slice ratios defined under every base station. Accordingly, bandwidth gets allocated as the calculated values, and the container gets initialized. There are two variables defined in this class, i.e.,

Capacity: It indicates the bandwidth allocated to the slice container.
Level: Indicates the container's current bandwidth level

```
def __init__(self, init, capacity):
```

This function initializes the data members of the container class.

Slice Module: This module features services that the client access. In other words, this class mimics real-world services. It checks whether the base station has the available bandwidth to allocate the resource to a client. It is because, for a service, we have a guaranteed bandwidth that needs to be fulfilled. If guaranteed bandwidth is not available, then the client should search for other nearby base stations. This function initializes the data member of the slice class.

Distributor Module: This function is responsible for the even distribution of clients to different slices. It also acts as a load balancer module in case of a compromise of any slice.

Stats Module: This class keeps track of the statistics associated with the base stations and the client.

```
def __init__(self, env, base_stations, clients, area):
```

This module initializes the data member of the stats class.

Graph Module: This module makes use of the Matplotlib library for plotting graphs. This function initializes the graph class data when called upon and draws the "Client and Base Station Graph", showing the base station and clients. Base stations are depicted as circles in different random colours, while clients are shown with different colour symbols. These symbols depict the type of service they have requested. All the symbols will be shown below the graph in the "Slice" sections. The area represented is 2km (nearly) shown by the x-axis and y-axis.

Utils Module: The utility module is also one of the other essential modules of this application. It contains the utilities required for the execution of the application. Other than connecting to the main module, it collects formatted data from the KDTree module and passes it to the client module for processing and the graph module for final data presentation.

KDTree Module: This function arranges the base station and clients in K-dimensions in the space.

```
def kdtree(clients, base_stations):
```

Client Module: The client class illustrates the real-world IoT endpoints that play the same role as the IoT devices in the real-world would do.

Program libraries Module: This module contains all the other generic program libraries required for its functioning.

Security Implementation

We identified three components for implementing security in this application to be used during the simulation, i.e., IoT devices, base station, and network slice. We identified many vulnerabilities already existing in these three components during our study. Details of these security issues are discussed in the literature review section of this paper.

Therefore, we decided to implement the security solutions in these three possible scenarios, i.e.:

Attack on or compromised IoT devices.
Attack on base stations.
Attack on network slice.

These three scenarios were identified based on the number of vulnerable intrusion points, as shown in Figure 10, which are also discussed in the study done by GSMA (2017a). Figure 10 shows the exact security methodology as has been implemented in this simulation. All three vulnerable points were disabled as soon as intrusion gets identified. Taking a few examples, in the first scenario, a car in Slice 2 was found to be vulnerable, which was disabled during the simulation.

Similarly, in the second scenario, a vulnerable point in Slice 3 was found compromised. As a result, the full slice was disabled, and this scenario might have an adverse impact where all the IoT clients in this slice will also get disconnected from the network. The next step could be to move devices to other slices based on their qualification. The last scenario is where an intruder attacks the whole base station. In this situation whole base station gets disabled; this is a significant impact on any network because all the slices and resources connected to it are now disabled. In this scenario, limited resources were moved to a nearby base station, based on network bandwidth and their service level agreement.

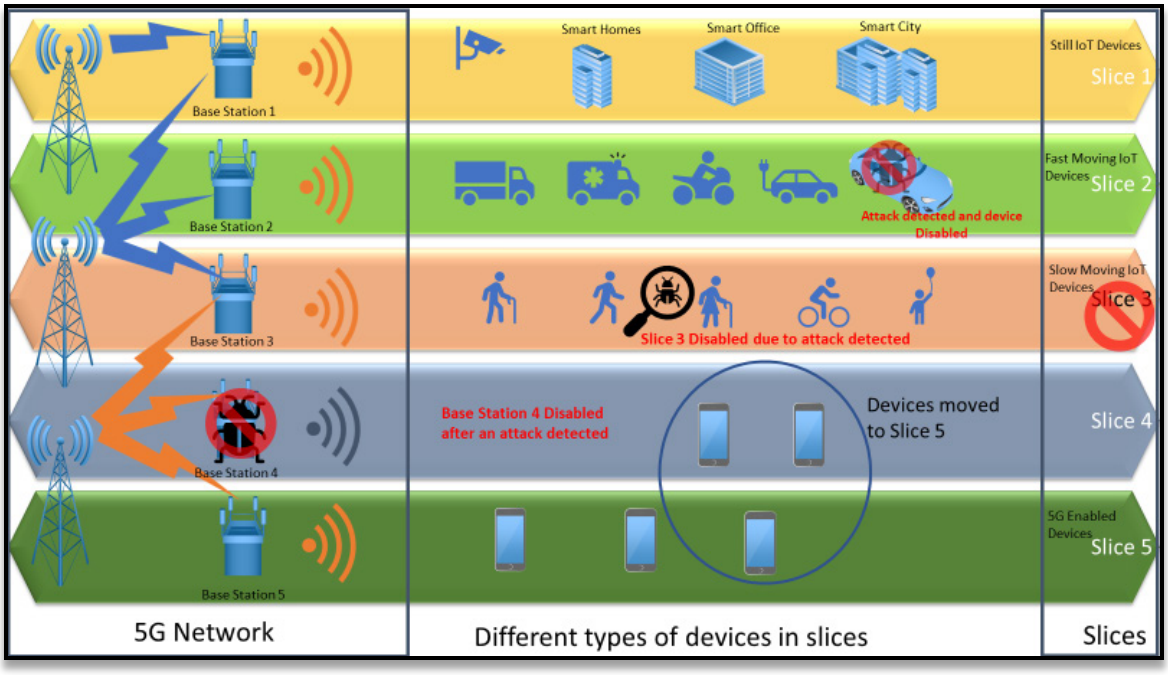

Figure 10. Simulated security implementation methodology on 5G enabled sliced network

Below two modes were designed to make the security implementation successful and effective. Software development best practices were followed while developing the methodologies of this application. All the modules developed were decoupled to ensure that each module knows very little about

each other, ensuring security and reducing each other's dependency. For the ease of handling, identification, and isolation of impacted resources, we divided this section of application into two separate modules:

Attack Mode: The target of this mode is to identify and isolate the attack on two components, i.e., IoT endpoints and base stations; this helps to contain the impact of the attack. This module is further divided into two parts, i.e., Level 1 and Level 2. Each level is designed to handle the attack on one resource, i.e., Level 1 for endpoints and Level 2 for base stations. All the compromised IoT devices or base stations identified here will be disabled, and their services will get stopped. As a result, this will help restrict further escalation of the attack and minimize its impact on the network and other resources.

Security Mode: Identification of compromised slice and disabling that slice will occur in this module. Impact or attack on any slice will get notified, and that slice will be immediately disabled to minimize and limit the impact to that slice only. Previously, the movement of impacted resources to other slices will depend on bandwidth availability and qualification.

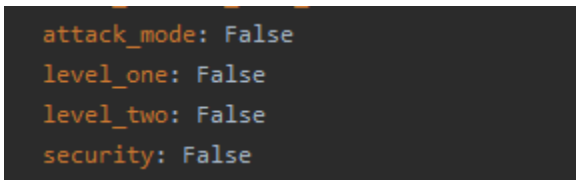

Figure 11. Security implementation setting

Along with other benefits mentioned above, all three resources were divided into three modules to build flexibility in this solution. As shown in Figure 11, all the modules can be easily enabled or disabled with a minor configuration change as per requirement, if any scenario requires one or the other module to be disabled or enabled. In this solution, we will be simulating all three possible scenarios to prove this feature's usability.

Intrusion Detection and Prevention Methodology

This simulation has used pattern matching or knowledge-based network IDS, as discussed in the previous section. Intrusion detection is one of the most challenging modules while developing any security solution because of modern attacks' varied nature. As seen in Figure 9, the security module is one of the subparts of the main module of the designed methodology. This module includes 16 parameters (hence called pattern matching or knowledge-based system) against which values will be matched, and it will act as an intrusion detection mechanism. These values are defined in a manually configurable configuration file, which can be changed as per requirement or change in a security scenario. Data defined against these values in a configuration file will be matched with the data set generated by the simulation network. If any mismatch in the data is found, an alert will get generated against that IoT device, base station, or slice. This alert will be forwarded to a subsection that will disable the compromised module.

These 16 configuration parameters were taken to validate the settings in this security solution.

```
#######attack detection settings

base_station_count=20
data_probing_time_sec=10
params_data_pushed=10
total_number_of_clients=5000
number_of_clients_per_slice=500
mobility_pattern=1,2,3,4,5
is_mobility_change_allowed=no
```

```
qos=1
change_client_weight=2
multiple_connection_request=4
high_mobilty=1
max_bw_usage=500
connection_ratio=1
handover_ratio=1
max_authentication_failure=5
connection_retry=10
```

Definition of a few configurations set as an example to understand the scenario:

max_authentication_failure=5, this setting checks the authentication failure count against any device. In case this counter becomes greater than 5, then that device will be disabled from using network resources.

number_of_clients_per_slice=500, here 500 has been set as a limit for the number of devices connected to any slice. In case it is found that the number of IoT devices connected to any slice is more than 500, the slice will be automatically blocked, and no new connections will be allowed.

max_bw_usage=500, in this simulation, we are using low powered and low data consuming IoT devices, and the data transmission limit is set at the device end. It is assumed that max data any device can transmit is 500 kbps. In case this limit is breached, we will assume that the device to be compromised. It can also be a part of a more significant DoS/DDoS attack, and as a result, that device will be disabled from using network resources.

params_data_pushed=10, this parameter is related to the one discussed above, but if an attacker manages to send compromised data within bandwidth limitation, it can be harmful to any network. Therefore, we have defined the limit on the number of parameters any device can send. In this case, it is set to Max 10. In case this limit is crossed, then the device is assumed to be compromised and, as a result, disabled.

These configurations can be further detailed, divided into many parts and subparts based on the client requirement. In case further enhancement is required, we also can include several artificial intelligence and machine learning techniques, which can provide a more advanced and real-time learning-based solution on the number of transactions happening in the network.

OUTPUT DATA

This simulation's output gets generated in two formats, i.e., graphical and textual (log file) data.

The Graphical Data consist of graphs that are expected from the simulation suite. These graphs will have calculated data that the simulation suite would plot. Textual data consists of the log data that each user would generate.

Figure 12 shows the sample graph generated during this testing simulation.

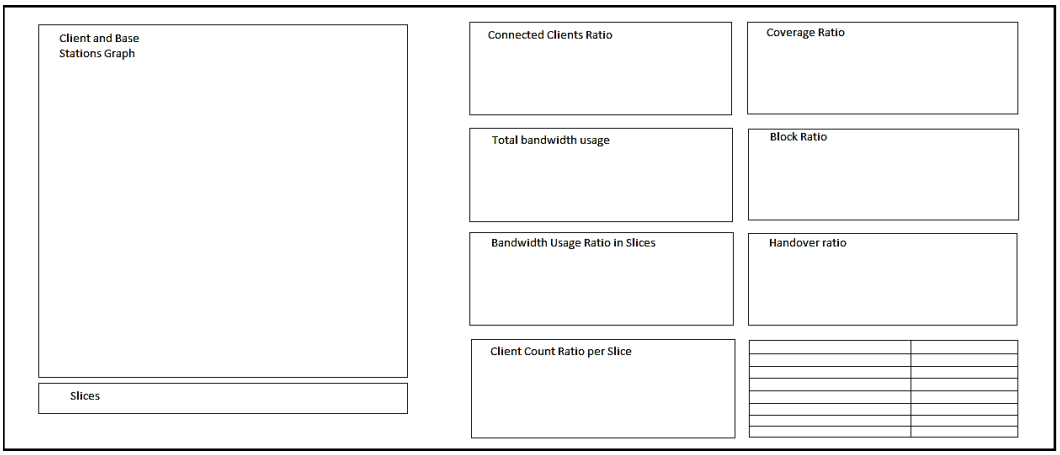

Figure 12. Output prototype of the graphs plotted

Below are the types of graphs generated.

Client and Base Stations Graph: This graph show clients and base stations. The area considered is around 2 km. Below are the ways to read and identify resources in this graph:

Circles represent base stations.
The slices section shows the details of the slices.
Clients and base stations are represented by the type of service it uses.

Connected Clients Ratio: The graph for the ratio of the number of clients connected to base stations to the total number of users represented against the time.

Total Bandwidth Usage: This graph represents the total bandwidth consumes by the clients against the time.

Coverage Ratio: The graph for the ratio of the area covered by the base stations to the total area under consideration represented against the time.

Block Ratio: The graph for the number of erroneous blocks' ratio to the total number of blocks sent represented against time.

Bandwidth Usage Ratio in Slices: The graph for the total bandwidth usage ratio in a slice to the total bandwidth allocated to that slice averaged for all slices represented against time.

Handover Ratio: The graph for the ratio of the number of clients moving from the coverage area of one base station to another to the total number of clients represented against time.

Client Count Ratio per Slice: The graph for the average number of people in a slice to be represented against time.

Logfile: Output here gets generated in the form of a text file. Contents of the log files are discussed below.

Base Stations Data: It shows the data of all the base stations that includes its ID, its location, coverage area, and the capacity of the base station (in bits).

Simulation Time: Starting with a timestamp of 1 with the details below.

Client location.
What service has the client requested to which base station?
Allocated bandwidth.
When the client puts back the service?
It also states scenarios such as when the client gets disconnected.

Data for individual client stating its, total connected time, total unconnected time, total request count, total consume time, total usage.

Data Collection and Verification

The data collection phase consists of identifying data sources, collecting data for the blocks (slices, mobility_pattern, base_stations, clients). Slices block name parameter consists of the name of the 5G New Radio (NR) use cases plus the voice service. 5G New Radio uses cases as discussed by X. Li et al. (2017) includes:

Enhanced Mobile Broadband (eMBB)
Massive Machine Type Communications (mMTC)
Ultra-Reliable Low Latency Communications (URLLC)
Prioritized Enhanced Mobile Broadband (eMBB_p)

Slices

Slices are the services that the client's request from the base station. Data collected for this consists of these parameters, guaranteed bandwidth, maximum bandwidth, and QoS class. Data utilized is displayed in Table 1.

Table 1. Data table for slices

Use Case	Guaranteed Bandwidth	Max Bandwidth	QoS Class
x_eMBB	-	100 Mbps	5
x_mMTC	1 Mbps	10 Mbps	2
x_URLLC	5 Mbps	10 Mbps	1
x_voice	500 Kbps	1 Mbps	3
y_eMBB	-	100 Mbps	5
y_eMBB_p	100 Mbps	10 Gbps	4
x_voice	500 Kbps	1 Mbps	3

IoT endpoints

A script written in Python generated the random data, i.e., IoT endpoints for the mobility pattern block. Client weight was calculated and assigned for all the clients, showing the ratio of IoT endpoints among cars, tram, walk, stationary, or slack. Further, their distribution and parameters were calculated. Data for this is shown in Table 2.

Table 2. Data table for mobility pattern

IoT Endpoints	Distribution	Parameters	Weight
Car	Normal Dist	N (μ=0, σ=7)	0.1
Walk	Random Integer	min=0, max=7	0.4
Stationary	Normal Dist	N (μ=0, σ=0.1)	0.2
Tram	Random Integer	min=-4, max=4	0.1
Slack	Random Integer	min=0, max=1	0.2

The program generated various types of IoT endpoints, which are selected randomly from the client list.

```
client_lst = [“stationary”, “car”, “walk”, “tram”, “slack”]
```

For the endpoints block, the parameter list included x and y, where x and y are used to specify the graph area in x and y directions. IoT device’s usage frequency and length of the area were specified, as mentioned in Table 3.

Table 3. Data table for client block

Name	Distribution	Parameters
X	Random Integer	Min=0, max=1980
Y	Random Integer	Min=0, max=1980
Usage Frequency	Random	Min=0, max=0.1

Base stations

Finding the data related to base stations was challenging. “Crowd-sourced cellular tower and coverage mapping service”, as shared by *Cellular Tower and Signal Map* (Cellmapper, 2020), was used to map a particular place’s cellular tower. Horwitz (2018) discussed the geographic information system of Fatih as the tools specially used for measurement. The “Fatih” area that we have examined consisted of around 17-23 small-to-large base stations defined by their coverage area. Data collected for this is shown in Table 4. The “Slices” parameter was to be defined as per the 5G use cases. Therefore, all the slices were included, and bandwidth was defined as per the randomly generated ratio.

Table 4. Data table for base stations

ID	Coverage (x,y),r (meters)	Throughput	Slices
1	(182,1414), 224	20 Gbps	[…]
…	…	…	…
5	(126,1016),384	30 Gbps	[…]
…	…	…	…
20	(44, 1916), 368	50 Gbps	[…]

Data collection and analysis resulted in the YAML file with values required for this simulation test.

We considered the below mandatory points while collecting and analyzing our data.

The area should have a high population and high mobility.
Data collected for an area of 2 km^2.
Data collected for the clients who are in the mentioned canvas area.
We ignored the clients outside the canvas area.
A circle represents base stations and locations taken to their real-time locations.
Considered a warmup and cooldown period of 5%, and data is captured only for those intervals.
The suite’s simulation time is taken as specified in the YAML file with 1 second as the time unit.

SIMULATION RESULTS

The simulation suite was tested for different values of "number_of_clients" with attack and security mode both as On and Off.

Tests simulated for different scenarios have been collated into result sets in the following scenarios.

1000, 2500, 5000 clients, with security and attack mode "OFF".
2500 clients, with security and attack mode "ON".

The time instance of every set is the same as 180 seconds.

RESULTSET OF 1000 CLIENTS WITH SECURITY AND ATTACK MODE OFF

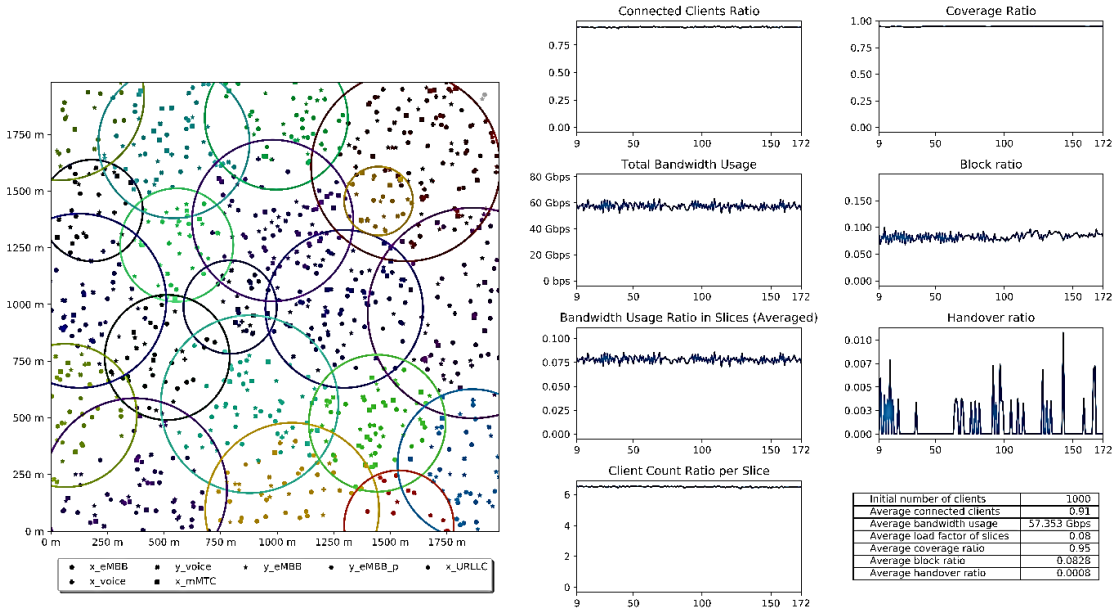

Figure 13. The graph plotted for 1000 clients

A network with few clients ought to have better connectivity than the same network with an increased number of clients. In the plotted graph, as shown in Figure 13, we can see the network has 91% active connections with block ratio being high and handover ratio being meagre.

This test was conducted to verify and ensure that all the network resources are functioning normally and as expected. During the simulation, both the security and attack mode was kept off, and therefore no intrusions were detected.

RESULTSET OF 2500 CLIENTS SECURITY AND ATTACK MODE OFF

The graph in Figure 14 depicts the scenario with an average number of clients, i.e., 2500, between 1000 (few) and 5000 (many). Such a network shows an optimal active connection and an optimal usage of the resources. The plotted graph shows that the network has 83% active connections with a moderately high block ratio, and the handover ratio is meagre.

This test was carried out with 2500 clients connected to ensure that all network resources are up with no adverse impact when load or number of active connections increases. In this simulation, both the security and attack modes were kept off, and therefore intrusion was not detected, keeping all resources are up and working as expected.

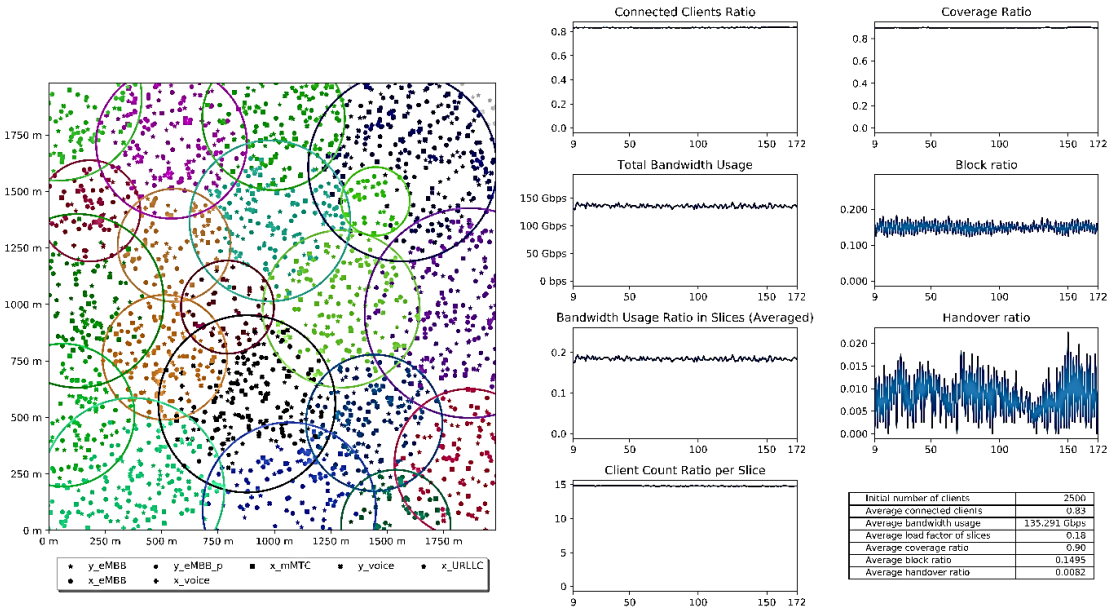

Figure 14. The graph plotted for 2500 clients

RESULTSET OF 2500 CLIENTS WITH ATTACK MODE ON

This test was conducted with the same number of clients as in the previous test exhibited with 2500 devices. The stability of the network and its resources are already tested, and no adverse impacts were detected. In this result set, attack mode was switched ON while keeping security mode is OFF, and therefore some intrusions were detected, keeping all other resources up and working as expected.

Figure 15 shows the input file changes to enable Level 1 and Level 2 of attack mode. These modes, as discussed in the previous section, will target IoT devices and base stations. Switching ON security mode is an optional setting that can be manually enabled to enhance the security profile.

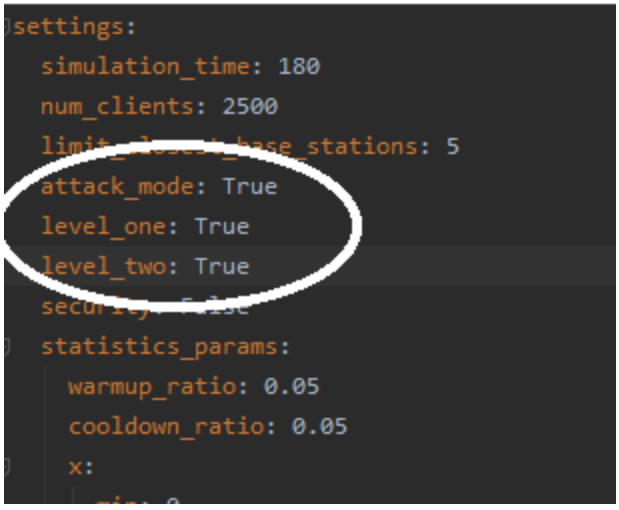

Figure 15. Attack mode enabled in input-file

Amidst this test, some of the settings which will impact base station and IoT clients were modified to create a compromised environment. As soon as the simulation detected these changes, they were

treated as an intrusion. As a result, the prevention mechanism got activated, which disabled impacted base stations, i.e., 13, 16, and 17.

As shown in Figure 16, text output and logs display three base stations 13, 16, and 17 are missing. These three base stations were disabled by the system for being detected as compromised. These base stations are now not allotting any resources to IoT devices.

```
BS_1     cov:[c=(182 , 1414), r= 224]    with cap 20000000000
BS_2     cov:[c=(556 , 1262), r= 250]    with cap 20000000000
BS_3     cov:[c=(514 ,  766), r= 276]    with cap 25000000000
BS_4     cov:[c=(64  ,  510), r= 316]    with cap 30000000000
BS_5     cov:[c=(126 , 1016), r= 384]    with cap 30000000000
BS_6     cov:[c=(1296,  980), r= 348]    with cap 25000000000
BS_7     cov:[c=(544 , 1714), r= 334]    with cap 25000000000
BS_8     cov:[c=(996 , 1822), r= 316]    with cap 30000000000
BS_9     cov:[c=(1568, 1608), r= 418]    with cap 80000000000
BS_10    cov:[c=(980 , 1370), r= 356]    with cap 35000000000
BS_11    cov:[c=(792 ,  988), r= 206]    with cap 35000000000
BS_12    cov:[c=(878 ,  560), r= 392]    with cap 40000000000
BS_14    cov:[c=(1066,   94), r= 384]    with cap 50000000000
BS_15    cov:[c=(1864,  962), r= 464]    with cap 100000000000
BS_18    cov:[c=(1538,   26), r= 242]    with cap 20000000000
BS_19    cov:[c=(1448, 1456), r= 152]    with cap 10000000000
BS_20    cov:[c=(44  , 1916), r= 368]    with cap 50
KDTREE CALL [0] - limit: 5
KDTREE CALL [1] - limit: 5
[1] Client_1 freshly assigned to BS_9    cov:[c=(1568, 1608), r= 418]    with cap 800000
[1] Client_2 freshly assigned to BS_12   cov:[c=(878 ,  560), r= 392]    with cap 400000
```

Base Station 13,16,17 missing

Figure 16. Log data containing an entry for base stations

Figure 17 shows the result in a graphical format. Base stations 13, 16, 17 were disabled after getting compromised. We can see the impacted sections highlighted in Figure 17, which shows both the base stations and IoT endpoints are now disabled.

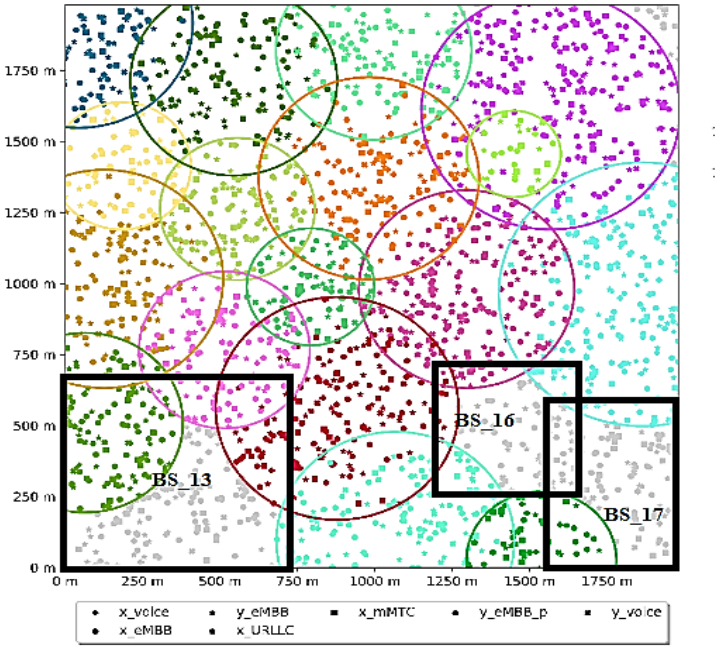

Figure 17. The graph plotted for attack mode

RESULTSET OF 2500 CLIENTS WITH SECURITY MODE ON

We conducted tests in this scenario with the same number of clients, i.e., 2500 devices. The stability of the network and its resource was found to be stable during this test as well. In this simulation, security mode was turned ON, which will detect impact or intrusion on slices. Attack mode was turned off, keeping all other resources up and working as expected.

As shown in Figure 18, this test was performed with security mode ON and the number of devices set to 2500.

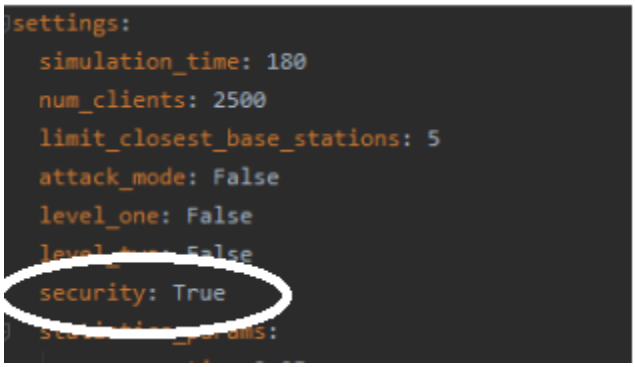

Figure 18. Changes made in input-file

As shown in Figure 19, logs show that the slice "y_embb" is connected with base Station 19. In this experiment, few settings were changed to simulate the intrusion scenario, impacting the slice. It can be seen in Figure 19 that slice "y_embb" was disabled, which restricted the escalation of intrusion to other slices. The obtained result confirms that our solution increases the effectiveness of network slicing in protecting resources. When a slice is compromised in a base station, it is necessary to bring down that slice, ensuring that no further resources are allocated. All the clients using that slice are migrated to other slices, or the service is unavailable. There can be many scenarios during a particular slice's disabling, such as migrating clients to the nearest available slice. The success of migration might depend upon the availability and approval from the nearest available and compatible slice. It also has to be considered that clients who are required to be moved are eligible, and they are not compromised. In a few scenarios, devices requesting movement can themselves be culprits. Therefore, deciding on a client's movement is not an easy task. It requires either manual intervention or smart machine learning techniques.

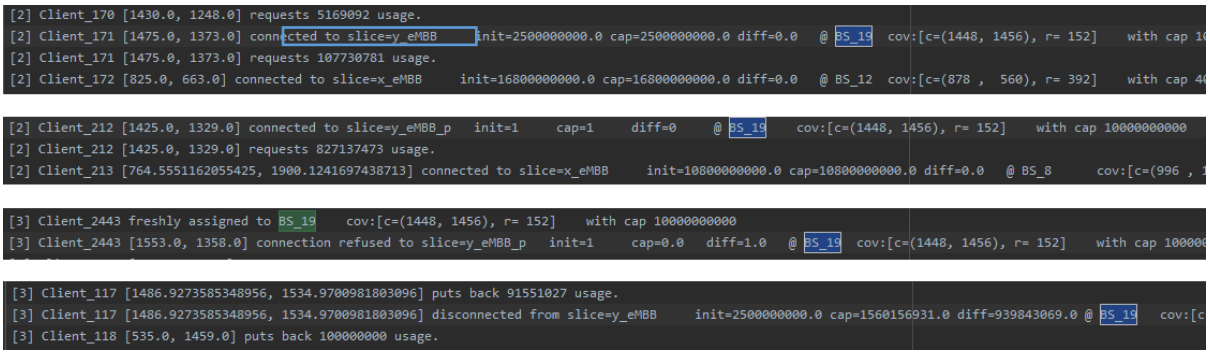

Figure 19. Log data

RESULTSET OF 5000 CLIENTS SECURITY AND ATTACK MODE OFF

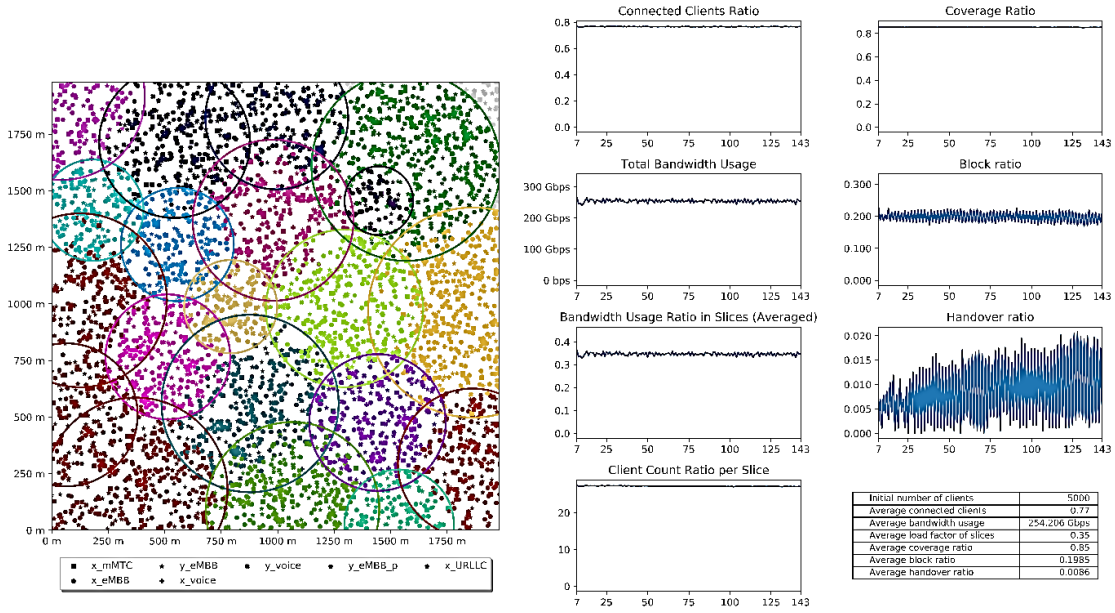

Figure 20. The graph plotted for 5000 devices

The test described in this section was finished with double the number of clients as done in the previous test with 2500 devices. This test ensures that all network resources are working fine with no adverse impact when the load or number of active connections increased. This simulation could not detect the intrusion because both the security and attack modes were OFF, and all resources were working fine. One more reason to conduct this test was to check the validity of results and establish a relation among all the test scenarios.

The graph in Figure 20 shows the network results with 5000 endpoints users. The number of active connections is less because there is an exhaustion of resources. Further, the allocation and deallocation of resources are very high. So, when one user is denied access to a resource, there may be an instance that resource is available for use after a few seconds. In the plotted graph, we can see the network has 77% active connections with block ratio almost zero and handover ratio very high.

COMPARATIVE ANALYSIS OF SIMULATION RESULTS

This simulation was performed by changing the number of clients and using other security and attack modes. The number of IoT endpoints was gradually increased from 1000 to 2500 and then to 5000, keeping the time instance constant to 180 seconds. As shown in Table 5, relations among the data generated were further analyzed, and we saw the relations confirming that the results were correct.

Table 5. Data collected from result sets

Parameter	1000 clients	2500 clients	5000 clients
Block	0.0828	0.0082	0
Handover	0.0008	0.1495	0.1985
Utilization	0.057	0.135	0.254
Connection	0.91	0.83	0.77

We need to note that security and attack mode was enabled only with 2500 IoT endpoints and not with 5000 devices because a test with 5000 endpoints devices was conducted only to establish a relation among conducted tests and confirm that the results are correct.

We have Block, Handover, Utilization, and Connection as the parameters for analyzing various result sets.

Block: It refers to the number of error blocks' ratio to the total number of blocks.

Handover: It refers to the ratio of handovers to the total number of clients in the network area.

Utilization: It is the amount of bandwidth consumed by the network divided by 1000.

Connection: It is the ratio of the number of active connections to the total number of users.

Plotting a graph for all these parameters gives a result, as shown in Figure 21.

Figure 21. Comparison graph

From Figure 21, we can make the following observations:

Connection: For n=5000, the number of active connections is less than n=1000, which has more active connections. Therefore, we can conclude that greater the number of clients in a network, the lower the chance of getting a connection whereas less the number of clients in a network, the more the chance of getting a connection.

Utilization: Utilization is a straight-forward concept where more people means utilization is high, fewer people means low utilization.

Handover: Handover happens when a user moves from one base station to the other. Therefore, many people mean more inter-movement between base-stations; fewer people means less inter-movement between base-stations.

Block: Block is an inverse concept; the more the number of people, the lower is the block error, while the fewer number of people will have more block error.

The above results show that the simulation tests performed are correct because the results generated are in accordance with the real-world model.

We have analyzed that security and attack mode worked as expected from the data generated after attack mode was enabled in the second scenario for connections with 2500 IoT devices.

In Attack Mode, we can see from the logs in Figure 16, and the graph in Figure 17 that base stations were disabled after discovering that specific endpoints or networks are compromised. The same can also be seen in the terminal output, as shown in Figure 22.

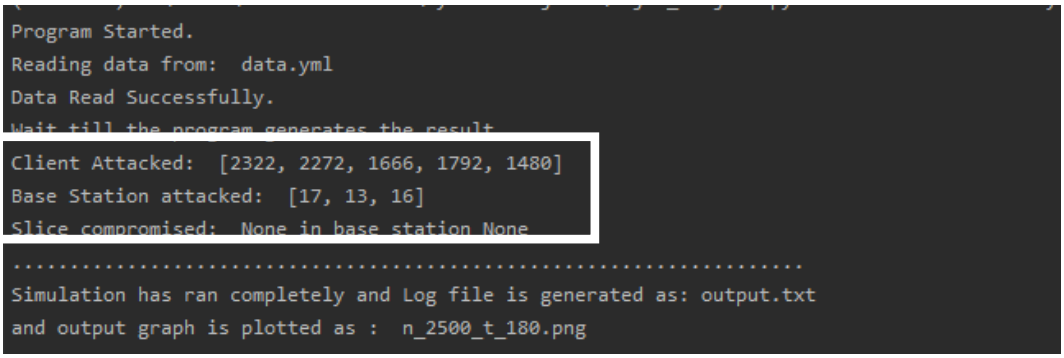

Figure 22. Command terminal confirming detection of the attack on client and base station

In Security Mode, logs in Figure 19 confirmed our solution's effectiveness in network slicing. Results show that the compromised slice was isolated, and the application did not allow new connections to ensure the slice's security. The same can also be seen in the terminal output, as shown in Figure 23.

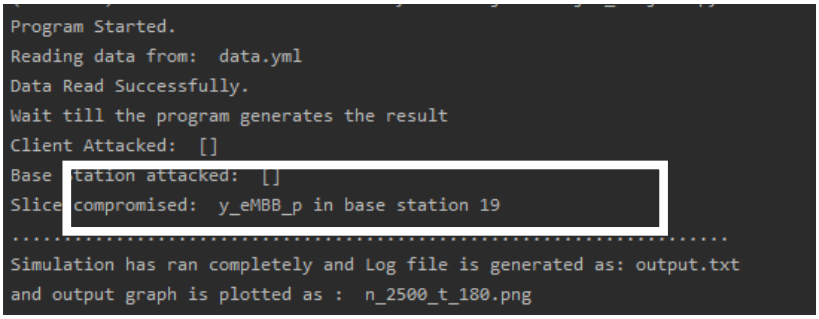

Figure 23. Command terminal confirming detection of the attack on a slice

We established a relation among all the tests performed during these whole testing and simulations, and the benefits of the security solutions implemented were exhibited. The intrusion detection module successfully detected the impacted resources like IoT endpoints, base stations, or network slices after the values were modified to generate an artificial attack. All the logs and graphs shown in previous sections show all the operations performed during this whole simulation.

CONCLUSION AND FUTURE WORK

From the result sets generated during this whole simulation process and from their analysis, we conclude that an IoT ecosystem's security increased by implementing the pattern matching IDS solution illustrated in this paper. The simulation here uses the same slices for customers with similar requirements, and our IDS solution helped to detect the intrusion on slices and base stations. This research's security module demonstrates a knowledge-based IDS solution, which can be modified and adopted as per the user's requirement to enhance security. When deployed over any network, this solution will increase the security with minimal impact on the network.

The increasing number of clients in a network increases networks utilization bandwidth, and hence the block ratio also rises. We can visualize the network utilization for the area by feeding in the correct data in the input-file. This simulation test can be used for analyzing mobility, usage pattern of IoT endpoints, effects of usage on frequency, and base station level impact of IoT devices or any intrusion on the network. This simulation suite's security module has successfully demonstrated its part when tested on 2500 devices, and several attack scenarios were tested. The intrusion was successfully detected, and impacted resources, i.e., IoT devices, base station, or a slice, was disabled as soon as it identified any mismatch or attack. We performed all the different scenarios starting from 1000, 2500 to 5000 to validate that the results aligned with real-world results. The same is visible from the output generated in graphical and text format.

Imaginations and technology enhancements have no limit; therefore, this simulation suit can also be enhanced to a new level and in different ways as per the user's requirement and improvement in technology. In this simulation, we expanded our previous research by Jain et al. (2020), where a manual simulation was performed with 10 IoT endpoints. Here, other than introducing technological advancements, such as IDS, endpoints were also increased to 5000 IoT devices. Practitioners can effectively use this simulation as an affordable solution to deploy over a sliced network.

Still, there is a way ahead; researchers can add more features to the main module for better visualization and analysis of base stations and IoT endpoints. Security implementation and intrusion detection and prevention module can be augmented by embedding artificial intelligence and machine learning techniques to use available data sets. Data can be generated as per requirement based on the transactions happening in the network. The manual configuration done in this simulation can be automated, which will make this module more independent and intelligent. The requirement in the network can change based on the vendor and organization it is using. Therefore, more modules can be added to accommodate requirements based on the client's usage and different scenarios that arise out of this need. Instead of directly disabling a device or resource, it can be moved to quarantine or red zone until it is declared safe by authorized sources to move it back to the network. We can also work on the automatic allocation of network slicing done manually in this simulation test. At last, future tests can be performed with a wide range of base stations and a large number (maybe millions) of IoT devices to analyze and capture a more significant challenge with built-in artificial intelligence techniques.

REFERENCES

3GPP. (2019). 3rd Generation partnership project, technical specification group services and system aspects, Release 15. *First 5G NR Specs Approved*. https://www.3gpp.org/news-events/1929-nsa_nr_5g

5G Americas. (2019). *The evolution of security in 5G - A "slice" of mobile threats*. https://www.5gamericas.org/wp-content/uploads/2019/08/5G-Security-White-Paper_8.15.pdf

Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A., & Flinck, H. (2018). Network slicing and softwarization: A survey on principles, enabling technologies, and solutions. *IEEE Communications Surveys & Tutorials*, *20*(3), 2429–2453. https://doi.org/10.1109/COMST.2018.2815638

Al-Bahri, M., Yankovsky, A., Borodin, A., & Kirichek, R. (2018). Testbed for identify IoT-devices based on digital object architecture. In O. Galinina, S. Andreev, S. Balandin, & Y. Koucheryavy (Eds), *Internet of things, smart spaces, and next generation networks and systems* (pp. 129–137). NEW2AN 2018, ruSMART 2018. *Lecture Notes in Computer Science*, vol. 11118. Springer, Cham. https://doi.org/10.1007/978-3-030-01168-0_12

Andrews, J. G., Buzzi, S., Choi, W., Hanly, S. V., Lozano, A., Soong, A. C. K., & Zhang, J. C. (2014). What will 5G be? *IEEE Journal on Selected Areas in Communications*, *32*(6), 1065–1082. https://doi.org/10.1109/JSAC.2014.2328098

Barakabitze, A. A., Ahmad, A., Mijumbi, R., & Hines, A. (2020). 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. *Computer Networks*, *167*, 106984. https://doi.org/10.1016/j.comnet.2019.106984

Becher, M., Freiling, F. C., Hoffmann, J., Holz, T., Uellenbeck, S., & Wolf, C. (2011). Mobile security catching up? Revealing the nuts and bolts of the security of mobile devices. *2011 IEEE Symposium on Security and Privacy*, 96–111. https://doi.org/10.1109/SP.2011.29

Boneh, D., Lynn, B., & Shacham, H. (2001). Short signatures from the weil pairing. In C. Boyd (Ed), *Advances in Cryptology* - ASIACRYPT 2001. *Lecture Notes in Computer Science*, vol 2248. Springer https://doi.org/10.1007/3-540-45682-1_30

Boussard, M., Bui, D. T., Douville, R., Justen, P., Sauze, N. Le, Peloso, P., Vandeputte, F., & Verdot, V. (2018). Future spaces: Reinventing the home network for better security and automation in the IoT era. *Sensors (Switzerland)*, *18*(9), 2986. https://doi.org/10.3390/s18092986

Boutigny, F., Betgé-Brezetz, S., Blanc, G., Lavignotte, A., Debar, H., & Jmila, H. (2020). Solving security constraints for 5G slice embedding: A proof-of-concept. *Computers & Security*, *89*, 101662. https://doi.org/10.1016/j.cose.2019.101662

Brahmi, N. (2013). *METIS: Mobile Communications for 2020 and beyond*. VDE/ITG Fachtagung Mobilkommunikation. https://metis2020.com/wp-content/uploads/publications/VDE_ITG_2013_Brahmi_Mobile_Communications-.pdf

CellMapper. (2020). *Cellular tower and signal map*. https://www.cellmapper.net/map

Chaabouni, N., Mosbah, M., Zemmari, A., & Sauvignac, C. (2020). A OneM2M intrusion detection and prevention system based on edge machine learning. *NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium*, 1–7. https://doi.org/10.1109/NOMS47738.2020.9110473

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., & Faruki, P. (2019). Network intrusion detection for IoT security based on learning techniques. *IEEE Communications Surveys & Tutorials*, *21*(3), 2671–2701. https://doi.org/10.1109/COMST.2019.2896380

Eskandari, M., Janjua, Z. H., Vecchio, M., & Antonelli, F. (2020). Passban IDS: An intelligent anomaly-based intrusion detection system for IoT edge devices. *IEEE Internet of Things Journal*, *7*(8), 6882–6897. https://doi.org/10.1109/JIOT.2020.2970501

Fang, Q., WeiJie, Z., Guojun, W., & Hui, F. (2014). Unified security architecture research for 5G wireless system. *2014 11th Web Information System and Application Conference*, 91–94. https://doi.org/10.1109/WISA.2014.25

Forsberg, D., Leping, H., Tsuyoshi, K., & Alanara, S. (2007). Enhancing security and privacy in 3GPP E-UTRAN radio interface. *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, 1–5. https://doi.org/10.1109/PIMRC.2007.4394792

Gentry, C. (2003). Certificate-based encryption and the certificate revocation problem. In E. Biham (Ed), *Advances in cryptology* - EUROCRYPT 2003. EUROCRYPT 2003. *Lecture Notes in Computer Science*, vol 2656. Springer. https://doi.org/10.1007/3-540-39200-9_17

Goldberg, R. P. (1974). Survey of virtual machine research. *Computer*, *7*(6), 34–45. https://doi.org/10.1109/MC.1974.6323581

GSMA. (2017a). An-introduction-to-network-slicing. https://www.gsma.com/futurenetworks/wp-content/uploads/2017/11/GSMA-An-Introduction-to-Network-Slicing.pdf

GSMA. (2017b). *Smart 5G networks: enabled by network slicing and tailored to customers' needs.* https://www.gsma.com/futurenetworks/wp-content/uploads/2017/09/5G-Network-Slicing-Report.pdf

Hedman, P. (Ed.). (2016). Description of network slicing concept. *Next Generation Mobile Networks (NGMN).* https://www.ngmn.org/wp-content/uploads/160113_NGMN_Network_Slicing_v1_0.pdf

Horwitz, J. (2018). *Decoding 5G: A cheat sheet for next-gen cellular concepts and jargon.* https://venturebeat.com/2018/12/12/decoding-5g-a-cheat-sheet-for-next-gen-cellular-concepts-and-jargon/

Jain, A., & Singh, T. (2019). Securing communication in IoT ecosystem using cryptographic algorithms. *International Journal of Engineering and Advanced Technology*, *9*(1), 7258–7268. https://doi.org/10.35940/ijeat.A1851.109119

Jain, A., & Singh, T. (2020) Security challenges and solutions of IoT ecosystem. In M. Tuba, S. Akashe, & A. Joshi (Eds), *Information and communication technology for sustainable development. Advances in intelligent systems and computing*, vol 933. Springer. https://doi.org/10.1007/978-981-13-7166-0_25

Jain, A., Singh, T., & Jain, N. (2021). Framework for securing IoT ecosystem using blockchain: Use cases suggesting theoretical architecture. In M. Tuba, S. Akashe, & A. Joshi (Eds), *CT systems and sustainability. Advances in intelligent systems and computing,* vol 1270. Springer. https://doi.org/10.1007/978-981-15-8289-9_21

Jain, A., Singh, T., Sharma, S. K., Marwaha, R., & Bardia, R. (2020). Securing IoT endpoints using network slicing on 5G network. *Solid State Technology, 63*(5), 689-705. http://solidstatetechnology.us/index.php/JSST/article/view/1615

Javaid, N., Sher, A., Nasir, H., & Guizani, N. (2018). Intelligence in IoT-based 5G networks: Opportunities and challenges. *IEEE Communications Magazine*, *56*(10), 94–100. https://doi.org/10.1109/MCOM.2018.1800036

Jover, R. (2013). Security attacks against the availability of LTE mobility networks: Overview and research directions. *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 1–9.

Khettab, Y., Bagaa, M., Dutra, D. L. C., Taleb, T., & Toumi, N. (2018). Virtual security as a service for 5G verticals. *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, 1–6. https://doi.org/10.1109/WCNC.2018.8377298

Kong, Q., Lu, R., Chen, S., & Zhu, H. (2016). Achieve secure handover session key management via mobile relay in LTE-advanced networks. *IEEE Internet of Things Journal*, *4*(1), 29-39. https://doi.org/10.1109/JIOT.2016.2614976

Kotulski, Z., Nowak, T. W., Sepczuk, M., & Tunia, M. A. (2018). Graph-based quantitative description of networks' slices isolation. In M. Ganzha, L. Maciaszek, * M. Paprzycki (Eds), *2018 Federated Conference on Computer Science and Information Systems*. ACSIS, Vol. 15, 369–379. https://doi.org/10.15439/2018F322

Kotulski, Z., Nowak, T. W., Sepczuk, M., & Tunia, M. A. (2020). 5G networks: Types of isolation and their parameters in RAN and CN slices. *Computer Networks*, *171*, 107135. https://doi.org/10.1016/j.comnet.2020.107135

Krishna, A. M., & Tyagi, A. K. (2020). Intrusion detection in intelligent transportation system and its applications using blockchain technology. *2020 International Conference on Emerging Trends in Information Technology and Engineering (Ic-ETITE)*, 1–8. https://doi.org/10.1109/ic-ETITE47903.2020.332

Ksentini, A., & Frangoudis, P. A. (2020). Toward slicing-enabled multi-access edge computing in 5G. *IEEE Network*, *34*(2), 99–105. https://doi.org/10.1109/MNET.001.1900261

Li, F., Peng, W., Huang, C.-T., & Zou, X. (2013). Smartphone strategic sampling in defending enterprise network security. *2013 IEEE International Conference on Communications (ICC)*, 2155–2159. https://doi.org/10.1109/ICC.2013.6654846

Li, X., Samaka, M., Chan, H. A., Bhamare, D., Gupta, L., Guo, C., & Jain, R. (2017). Network slicing for 5G: Challenges and opportunities. *IEEE Internet Computing*, *21*(5), 20–27. https://doi.org/10.1109/MIC.2017.3481355

Liu, Q., Han, T., & Ansari, N. (2020). Learning-assisted secure end-to-end network slicing for cyber-physical systems. *IEEE Network*, *34*(3), 37–43. https://doi.org/10.1109/MNET.011.1900303

Mahajan, S., & Jindal, P. A. (2010). Security and privacy in VANET to reduce authentication overhead for rapid roaming networks. *International Journal of Computer Applications*, *1*(20), 21–25. https://doi.org/10.5120/428-631

Mantas, G., Komninos, N., Rodriuez, J., Logota, E., & Marques, H. (2015). Security for 5G communications. In J. Rodriguez (Ed.), *Fundamentals of 5G mobile networks* (pp. 207–220). John Wiley & Sons. https://doi.org/10.1002/9781118867464.ch9

Martini, B., Mori, P., Marino, F., Saracino, A., Lunardelli, A., Marra, A. La, Martinelli, F., & Castoldi, P. (2020). Pushing forward security in network slicing by leveraging continuous usage control. *IEEE Communications Magazine*, *58*(7), 65–71. https://doi.org/10.1109/MCOM.001.1900712

Mathew, A. (2020). Network slicing in 5G and the security concerns. *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 75–78. https://doi.org/10.1109/ICCMC48092.2020.ICCMC-00014

Modi, C., Patel, D., Borisaniya, B., Patel, A., & Rajarajan, M. (2013). A survey on security issues and solutions at different layers of cloud computing. *The Journal of Supercomputing, 63*, 561–592. https://doi.org/10.1007/s11227-012-0831-5

Nakao, A., Du, P., Kiriha, Y., Granelli, F., Gebremariam, A. A., Taleb, T., & Bagaa, M. (2017). End-to-end network slicing for 5G mobile networks. *Journal of Information Processing*, *25*, 153–163. https://doi.org/10.2197/ipsjjip.25.153

Nanda, S., & Tzi-cker, C. (2005). *A survey on virtualization technologies*. https://rtcl.eecs.umich.edu/papers/publications/2011/TR179.pdf

Next Generation Mobile Networks (NGMN). (2020). *Pushing the 5G ecosystem*. https://www.ngmn.org/ngmn-news/press-release/ngmn-alliance-publishes-second-5g-white-paper.html

Ni, J., Lin, X., & Shen, X. S. (2018). Efficient and secure service-oriented authentication supporting network slicing for 5G-enabled IoT. *IEEE Journal on Selected Areas in Communications*, *36*(3), 644–657. https://doi.org/10.1109/JSAC.2018.2815418

Ordonez-Lucena, J., Ameigeiras, P., Lopez, D., Ramos-Munoz, J. J., Lorca, J., & Folgueira, J. (2017). Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges. *IEEE Communications Magazine*, *55*(5), 80–87. https://doi.org/10.1109/MCOM.2017.1600935

Oshin, O., Luka, M., & Atayero, A. (2016). From 3GPP LTE to 5G: An evolution. In S. Ao, G-C. Yang, & L. Gelman (Eds), *Transactions on engineering technologies*, 485–502. Springer. https://doi.org/10.1007/978-981-10-1088-0_36

Pirinen, P. (2014). A brief overview of 5G research activities. *1st International Conference on 5G for Ubiquitous Connectivity, 5GU, IEEE*. https://doi.org/10.4108/icst.5gu.2014.258061

Rost, P., Mannweiler, C., Michalopoulos, D. S., Sartori, C., Sciancalepore, V., Sastry, N., Holland, O., Tayade, S., Han, B., Bega, D., Aziz, D., & Bakker, H. (2017). Network slicing to enable scalability and flexibility in 5G mobile networks. *IEEE Communications Magazine*, *55*(5), 72–79. https://doi.org/10.1109/MCOM.2017.1600920

Sathi, V. N., Srinivasan, M., Kaliyammal Thiruvasagam, P., & Murthy, S. R. (2020). Novel protocols to mitigate network slice topology learning attacks and protect privacy of users' service access behavior in softwarized 5G networks. *IEEE Transactions on Dependable and Secure Computing*. https://doi.org/10.1109/TDSC.2020.2968885

Sattar, D., & Matrawy, A. (2019). Towards secure slicing: Using slice isolation to mitigate DDoS attacks on 5G core network slices. *2019 IEEE Conference on Communications and Network Security (CNS)*, 82–90. https://doi.org/10.1109/CNS.2019.8802852

Saxena, N., Grijalva, S., & Chaudhari, N. S. (2016). Authentication protocol for an IoT-enabled LTE network. *ACM Transactions on Internet Technology*, *16*(4), 1–20. https://doi.org/10.1145/2981547

Scarfone, K., & Mell, P. (2007). *Guide to intrusion detection and prevention systems (IDPS)*. Computer Security Resource Center, NIST. https://csrc.nist.gov/publications/detail/sp/800-94/final

Schinianakis, D. (2017). Alternative security options in the 5G and IoT era. *IEEE Circuits and Systems Magazine*, *17*(4), 6–28. https://doi.org/10.1109/MCAS.2017.2757080

Sciancalepore, V., Zanzi, L., Costa-Perez, X., & Capone, A. (2018). *ONETS: Online network slice broker from theory to practice*. http://arxiv.org/abs/1801.03484

Seddigh, N., Nandy, B., Makkar, R., & Beaumont, J. F. (2010). Security advances and challenges in 4G wireless networks. *2010 Eighth International Conference on Privacy, Security and Trust*, 62–71. https://doi.org/10.1109/PST.2010.5593244

Sekar, R., Guang, Y., Verma, S., & Shanbhag, T. (1999). A high-performance network intrusion detection system. *Proceedings of the 6th ACM Conference on Computer and Communications Security - CCS '99*, 8–17. https://doi.org/10.1145/319709.319712

Shamir, A. (2000). Identity-based cryptosystems and signature schemes. In G.R. Blakley & D. Chaum (Eds), *Advances in Cryptology*. CRYPTO 1984. *Lecture Notes in Computer Science,* vol 196, (pp. 47–53). Springer. https://doi.org/10.1007/3-540-39568-7_5

Singh, T., Verma, S., Kulshrestha, V., & Katiyar, S. (2016). Intrusion detection system using genetic algorithm for cloud. *Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies - ICTCS '16*, 1–6. https://doi.org/10.1145/2905055.2905175

Singh, T., Verma, S., & Parashar, V. (2016). Securing internet of things in 5G using audio steganography. In A. Unal, M. Nayak, D. Mishra, D. Singh, & A. Joshi (Eds), *Smart Trends in Information Technology and Computer Communications*. SmartCom 2016. *Communications in Computer and Information Science*, vol 628, (pp. 365–372). Springer. https://doi.org/10.1007/978-981-10-3433-6_44

Sun, S., Gong, L., Rong, B., & Lu, K. (2015). An intelligent SDN framework for 5G heterogeneous networks. *IEEE Communications Magazine*, *53*(11), 142–147. https://doi.org/10.1109/MCOM.2015.7321983

Togou, M. A., Bi, T., Dev, K., McDonnell, K., Milenovic, A., Tewari, H., & Muntean, G.-M. (2020). DBNS: A distributed blockchain-enabled network slicing framework for 5G networks. *IEEE Communications Magazine*, *58*(11), 90–96. https://doi.org/10.1109/MCOM.001.2000112

Tutorials Point. (2020). *Learn 5G*. https://www.tutorialspoint.com/5g/index.htm

Wang, D., Chen, D., Song, B., Guizani, N., Yu, X., & Du, X. (2018). From IoT to 5G I-IoT: The next generation IoT-based intelligent algorithms and 5G technologies. *IEEE Communications Magazine*, *56*(10), 114–120. https://doi.org/10.1109/MCOM.2018.1701310

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., & de Alvarenga, S. C. (2017). A survey of intrusion detection in internet of things. *Journal of Network and Computer Applications*, *84*, 25–37. https://doi.org/10.1016/j.jnca.2017.02.009

Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., Pan, L., Maharjan, S., & Zhang, Y. (2016). Energy-efficient offloading for mobile edge computing in 5G heterogeneous networks. *IEEE Access*, *4*, 5896–5907. https://doi.org/10.1109/ACCESS.2016.2597169

AUTHORS

Anshul Jain received his Master's in Network Technology and Management from Amity University, Uttar Pradesh, India. He is currently pursuing his Ph. D. from Amity Institute of Information Technology, Amity University, U.P., India. He has a broad work experience of over 14 years in different multinational organizations in information security, integration, maintaining 5G and other Telecom software like VAS (VMS/SMSC/Prepaid Recharge/Mobile Money), IoT, Radio Frequency, Mobile Financial Services, VoIP. His interest and research areas include Information Security, Internet of Things, Blockchain, Artificial Intelligence. Currently, he is working as a Technical Manager in Amdocs.

Tanya Singh received her Bachelor's degree in Electronics from MIT, Dr. Babasaheb Ambedkar University, India. Master's in Information Technology from School of Information Technology, Guru Gobind Singh Indraprastha University, New Delhi, India and PhD in IT and Engineering from Banasthali University, India. Prof. (Dr.) Tanya Singh is currently working as Dy. Director (Academics), Amity University, Uttar Pradesh, India. With a comprehensive experience of over 20 years in Academia, Research, Planning, and Development in Education and Operational role outs, she has emerged as a Technical Evangelist for Networking and Cyber Security. Her aim is teaching and learning new technologies, assist and develop the knowledge amongst students, instructors by encouraging critical thinking, implementation, and ability to convert technological ideas into innovation. She has more than 40 research papers in reputed journals with Thompson ISI, Scopus.

Satyendra Kumar Sharma received his Bachelor's in Science in 1985 from CCS University, India, and a second Bachelor's in Computer Science in 1990 from Marathwada University, Aurangabad, India. He completed his Masters in Computer Science in 1993 from Roorkee University, India, and MBA in 2013 from Sikkim Manipal University, India. His first PhD was completed in Operational Research in 2011 from CCS University, India, and his second PhD in IT from J R N Rajasthan Vidyapeeth (Deemed) University, India. He is an academically enriched Educationist with rich academic and administrative experience, offering over an overall experience of 25+ years in various Engineering Colleges. He is an accomplished trainer and a mentor with a record of creative scholastic achievements. He has vast experience in education and research, having authored five technical books, around 100 research papers in different Journals and International and National Conferences.

Vikas Prajapati has completed his Bachelor of Technology in Computer Science and Engineering from Amity School of Engineering and Technology, Amity University, Uttar Pradesh, India. He has work experience in the field of Software Development. Currently, he is working as a Software Engineer. His research areas include cybersecurity, telecom, 5G, scripting, artificial intelligence, ethical hacking.