Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.043534

ARTICLE

Intrusion Detection Model Using Chaotic MAP for Network Coding Enabled Mobile Small Cells

Chanumolu Kiran Kumar and Nandhakumar Ramachandran*

School of Computer Science and Engineering, VIT-AP University, Amaravati, Andhra Pradesh, 522241, India

*Corresponding Author: Nandhakumar Ramachandran. Email: nandhakumarr03@gmail.com

Received: 05 July 2023 Accepted: 09 December 2023 Published: 26 March 2024

ABSTRACT

Wireless Network security management is difficult because of the ever-increasing number of wireless network malfunctions, vulnerabilities, and assaults. Complex security systems, such as Intrusion Detection Systems (IDS), are essential due to the limitations of simpler security measures, such as cryptography and firewalls. Due to their compact nature and low energy reserves, wireless networks present a significant challenge for security procedures. The features of small cells can cause threats to the network. Network Coding (NC) enabled small cells are vulnerable to various types of attacks. Avoiding attacks and performing secure "peer" to "peer" data transmission is a challenging task in small cells. Due to the low power and memory requirements of the proposed model, it is well suited to use with constrained small cells. An attacker cannot change the contents of data and generate a new Hashed Homomorphic Message Authentication Code (HHMAC) hash between transmissions since the HMAC function is generated using the shared secret. In this research, a chaotic sequence mapping based low overhead 1D Improved Logistic Map is used to secure "peer" to "peer" data transmission model using lightweight H-MAC (1D-LM-P2P-LHHMAC) is proposed with accurate intrusion detection. The proposed model is evaluated with the traditional models by considering various evaluation metrics like Vector Set Generation Accuracy Levels, Key Pair Generation Time Levels, Chaotic Map Accuracy Levels, Intrusion Detection Accuracy Levels, and the results represent that the proposed model performance in chaotic map accuracy level is 98% and intrusion detection is 98.2%. The proposed model is compared with the traditional models and the results represent that the proposed model secure data transmission levels are high.

KEYWORDS

Network coding; small cells; data transmission; intrusion detection model; hashed message authentication code; chaotic sequence mapping; secure transmission

1 Introduction

Mobile small cells enabled by Network Coding (NC) are seen as a viable technology for fifth-generation (5G) networks. 5G networks can span the metropolitan environment by being set up on demand anywhere at any time. Despite the many advantages this technology offers for 5G mobile networks, serious security concerns exist since NC-enabled mobile small cells are vulnerable to numerous attacks [1]. NC-enabled mobile small cells can function to their full capacity, and intrusion detection and prevention systems [2] are used to detect and neutralize several attacks.

Since small cells can enable efficient delivery of ubiquitous 5G services in a cost and energy-efficient way, they are viewed as a key 5G enabling technology. These nodes serve as mini-base stations, greatly benefiting network users [3]. Users of the network benefit from higher data rates and lower energy consumption and latency. The power of a radio signal decreases quadratically with its distance [4], therefore substituting long broadcasts to and from the Base Station (BS) with several shorter transmissions results in significant energy savings and minimizes the amount of interfering radio signals. The cost savings in electricity might be put toward enabling faster data transfers [5]. In addition, the latency is decreased since the physical transmission distance is cut in half when the source node and the destination node are near together.

It is anticipated that the advent of 5G mobile communications will lead to a more interconnected world. Small cell technology is a key enabler of 5G, allowing for the efficient and low-cost delivery of 5G services [6]. It is demonstrated that NC technology is a viable option for boosting throughput and bettering network performance in mobile small cells due to power usage, packet loss, and low network connectivity [7]. Despite NC's many advantages, wireless networks that employ the technology are vulnerable to attacks like pollution attacks. This is an attack in which a hostile node injects corrupted network packets, rendering the destination nodes incapable of decoding the native packets correctly [8]. An enormous amount of network resources and node energy are wasted during this damaging attack [9]. Given the significance of security in the rollout of 5G technology, a unique intrusion detection approach against these threats in the NC-enabled mobile small cells is necessary [10].

Many studies' primary goal was centered on developing methods for identifying these intrusions [11]. The preventative measures need to include something that stops these bad actors from polluting future packets [12]. It is crucial to pinpoint the position of a malicious user alongside detecting a security attack so that other nodes in the network can be alerted to the existence of an opponent [13]. Since network coding is used to create new energy-efficient and high-speed networking architectures for mobile small cells [14], it is important to deal with security attacks like pollution attacks that can severely degrade network performance and waste energy [15]. The suggested approach in this research aims to defend NC-enabled mobile small cells from pollution attacks using a lightweight HHMAC using achaotic function that might deplete resources including bandwidth and battery life [16].

Decentralizing software engineering tendencies met with existing networking technology to give birth to the idea of Peer-to-Peer (P2P) computing, which has been there since the early days of networking. One definition of a P2P paradigm displaces centralized computing in favor of a variant of client-server architecture [17]. This research defined dangerous P2P applications [18] as those that pose a risk to the safety of wired networks and characterized the traffic generated by these programs as malicious [19]. An intrusion is the unauthorized access to, or use of, a computer system or its resources with the intent to harm. To detect intrusion, one must be able to single out the people or computers responsible for it. To detect intrusions, IDS compares observed behavior to known malicious patterns, ideally in real time. To a large extent, intrusion is a network-based phenomenon. As the number of interconnected devices in the world grows, the issue of intrusion has come to the forefront, resulting in vigorous study of effective IDS.

Even while P2P networks are not initially imagined as dangerous software, some of their users may develop criminal intentions. Since the use of these applications varies, it is challenging for software suppliers to create security solutions to secure cooperation against them [20]. To protect a network host from the vulnerabilities presented by P2P applications, a novel approach is required due to the wide variety of purposes served by these programs. Since the complexity of random detector-generating methods rises exponentially with the size of the self-set scale, the chaos theory-based detector-generating algorithm provides a workaround. Although the property of algorithms has

improved to some degree, there are still issues with large search blinds and slow convergence speed. P2P networking is a network of computers in which each participant has the same level of access and is responsible for the same amount of data processing. Unlike traditional client-server networks, where a single device takes on both roles of data server and data client, P2P networks have no such hierarchy. The P2P intrusion detection model is shown in Fig. 1. The P2P model provides a link between different remote users to establish a connection using a Virtual Private Network (VPN) network. The firewall is established for intrusion detection between remote users. The intrusion response system helps in the identification of intrusions when connected to remote servers.

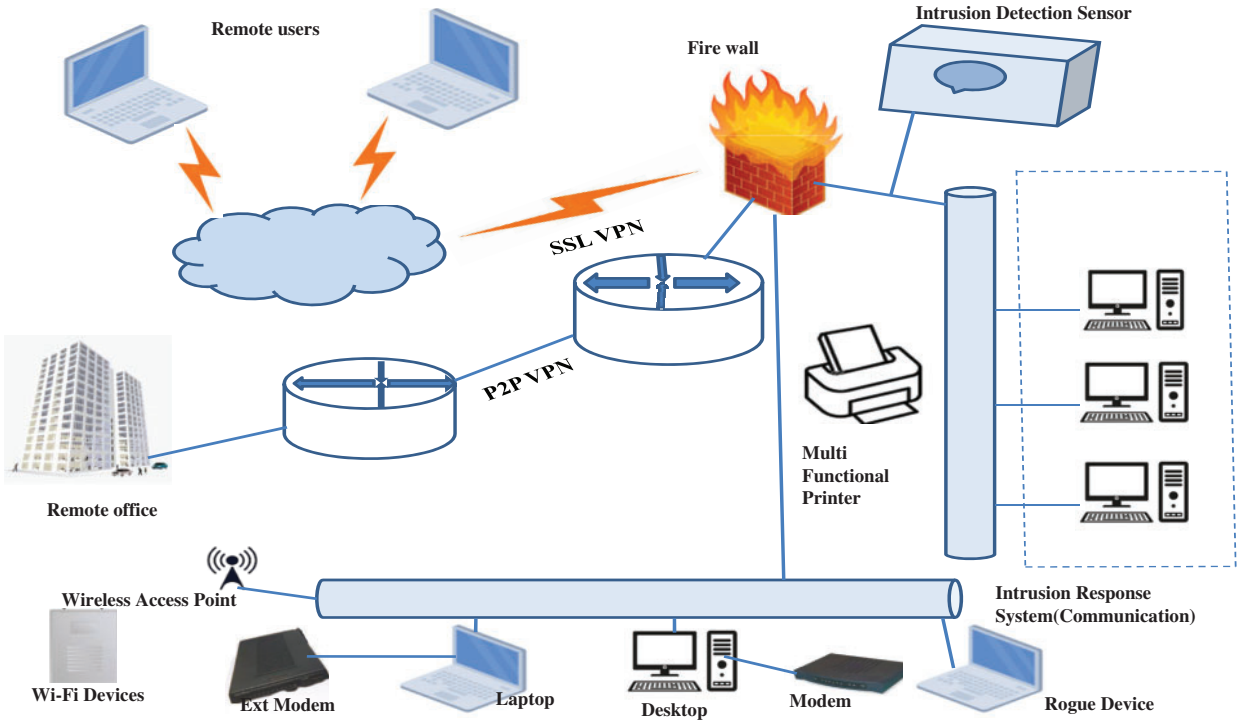

Figure 1: P2P intrusion detection model

Hash functions, symmetric ciphers, and asymmetric ciphers are examples of basic cryptographic approaches that need to be strengthened, making advancements in the field of cryptography essential. There are high hopes for its safety features because of the technology's shift to smaller, more portable designs. So, naturally, there is a subfield called lightweight cryptographic hash functions for use in a wide variety of contexts. A lightweight cryptographic hash designer must balance the competing needs for security, efficiency, and affordability [21]. One can choose from a variety of lightweight hash functions with differing degrees of security, each tailored to a certain set of needs or criteria [22].

Cryptography's hash function takes in packets of varying lengths and produces ones with a set length. The output of a hash function is a hash value, often known as a message digest [23] or fingerprint. Cryptographic hash functions [24] are used in a wide variety of contexts, including data integrity, entity authentication, digital signatures, pseudorandom number generators, key generation, password security, and blockchains. If the sender has a private key, they can use it to sign the message's hash value. Digital signatures rely heavily on the integrity of the underlying cryptographic hash function. The general hashing process is shown in Fig. 2.

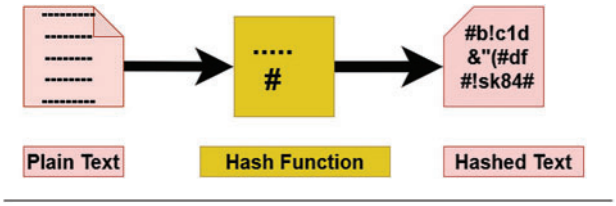

Figure 2: General hash process

Network coding has historically relied heavily on authentication systems based on HHMACs, with or without a homomorphic cryptographic signature (HCS). Verification in NC-compatible networks is expanding beyond fault tolerance. Current HMAC-based authentication techniques for NC-supporting networks not only drop malformed packets but also identify the malicious nodes that initiated the attack [25]. Nodes that detect corrupted packets must also have the capability to repair them if an NC-enabled network is to operate at peak efficiency using an intrusion model. Throughput is increased, and the communication cost related to re-transmission is reduced in this research. Different HMAC models are available but their limitations are overcome in this research. Selecting a suitable cryptographic hash function for the HMAC is essential for the operation of conventional models. SHA-1, SHA-256, and SHA-512 are popular options. The traditional model process is to pick a key to keep under wraps for the HMAC. The secret key should be a completely made-up, one-of-a-kind number that no one but the intended recipients knows. Users should provide the HMAC function with the message to the hash, the secret key to use, the length of the hash result, and the hashing strategy they prefer. Using the message, the secret key, and any additional information, the HMAC algorithm will generate a unique hash value. The HMAC function then returns the hash value, which can be used for message authentication. Fig. 3 shows the polluted packet transmission if proper authentication is not performed.

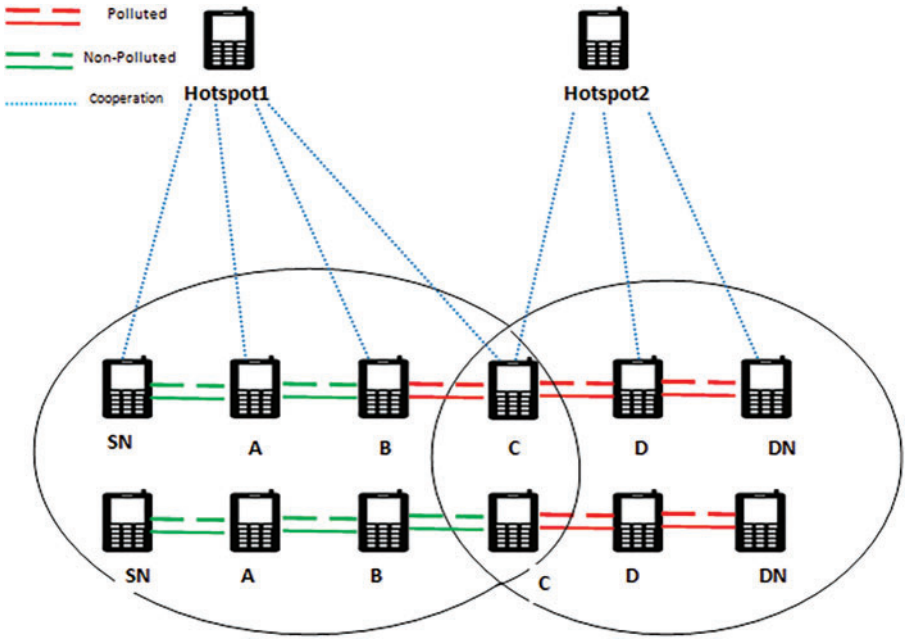

Figure 3: Polluted packet transmission

HMACs are widely used in NC authentication methods due to their homomorphic character and significantly lower computational overhead. There are two common strategies for implementing HMACs in new NC authentication methods. These are similar to HMACs in the null and subspace spaces. Authentication parameters in subspace HMACs are created to lie within the subspace formed by the payload of the packet [26]. In contrast, keys used to create authentication parameters on null space-based HMACs lie in a plane that is orthogonal to the subspace produced by the vectors they authenticate. A vector of symbols is used as the authenticating parameter instead of a single symbol. Keys are generated and confirmed by using unique key selection and distribution methods. Based on the homomorphic MAC used, the P2P-based communication is initiated and the intrusion detection model is activated during the transmission.

The term hash is shorthand for the output of a cryptographic hash function, which is a technique that accepts an arbitrary amount of data as input and returns an encoded string of a defined size. This encrypted string can then be saved in place of the actual password for use in node authentication. Password storage security can be affected by the characteristics of cryptographic hash functions. It will be very difficult to recover the original password from a strong hash. Using hash functions makes authentication more difficult, which helps prevent malicious actors from breaching a network. Hash functions are among the most popular cryptographic methods utilized today. Hash functions, also known as digest functions, are mathematical operations that take any data as input and return a string of predetermined length. Among the many security-related uses for hash functions are tests for data integrity, message authentication, and the production of keys. Using a secret key known only to the sender and receiver, the message can be hashed. A MAC, or message authentication code, is generated from this hash and attached to the message. After receiving the MAC, the receiver can compute a hash with the same key to verify its authenticity. If they are consistent, the communication can be trusted as genuine and unaltered. Since both sides are authenticated using the same key, this technique is also known as symmetric-key authentication. The types of hash functions are discussed.

1.1 Salted Hashes

Each plaintext credential is "salted" with a series of random bytes. As a result, two identical plaintext passwords are now distinguishable in the encrypted form.

1.2 Hash Functions with Keys

A keyed hash function, also known as HHMAC, is an algorithm for generating a keyed and hashed message authentication code using a cryptographic key and a cryptographic hash function.

1.3 Keyed Hash Functions

Any function that is intended to iterate on its inner workings, feeding the output back as input, will be considered an adaptive one-way function, even if doing so slows down the function's execution time. It is adaptable because the number of iterations can be changed on the fly by the programmer. Adaptive design has been implemented in hash algorithms and encryption techniques (like bcrypt) to safeguard saved passwords.

When given an initial seed value, chaotic maps are mathematical functions that produce a seemingly random pattern [27]. This paper summarizes the usage of chaotic maps in NC-enabled small cells. While the mathematical model behind a discrete chaos map is straightforward to implement and compute, the cells' parameter range is typically insufficient, and the wrong selection of parameters can rapidly degrade the system's dynamic characteristics.

Secure data transfer and intrusion detection in NC-enabled small cells have both benefited from applications of chaos theory's computational power. It has been demonstrated that some low-dimensional maps can be predicted with high accuracy because chaotic space is finite. However, high-dimensional maps have a considerably bigger chaotic space. The difficulty of implementing them effectively limits their utility for real-time intrusion detection. This research presents a novel fractional one-dimensional chaotic map that is well-suited to a space of this size. The proposed map is highly chaotic over a wide range of values for its control parameters, despite its relatively straightforward design. In this research, a new paradigm of private P2P data transfer for small cells is presented, which enables sensitive intrusion monitoring. Recent quantification studies suggest the existence of important extracted features in packet traffic with a more unique characteristic of fractal processes than traditional stochastic processes. A method for an asymmetric key block cipher is proposed that makes use of a set of chaotic maps in one dimension rather than a single map of that dimension. In this research, a lightweight hash H-MAC-based chaotic sequence mapping-based secure P2P data transmission paradigm is proposed with precise intrusion detection.

2 Literature Survey

A lightweight protocol was employed by Siddharthan et al. [1] to manage the time-constrained issue, and an intelligent system for intrusion detection was designed to recognize or forecast a cyber-attack utilizing Elite algorithms for machine learning (EML). All of the hardware and sensors are connected using the lightweight Message Queue Telemetry Transport (MQTT) protocol to do the experimental analysis of the work on a test bed environment. The SEN-MQTTSET dataset was collected from three different scenarios with the help of sensors, multi-context features were generated from the dataset using an ensemble mathematical multi-view cascade generation of features algorithm, and the dataset was evaluated using ML algorithms. From these three cases, normal, subscriber assault, and broker attack, the SEN-MQTTSET dataset was generated. The raw dataset was sent through an ensemble analytical multi-view cascade development of features algorithm, which produces the multi-context feature.

Network security relies heavily on the timely identification of attack attempts. It is challenging to detect assaults before a session finishes since most studies of network IDS systems use features for whole sessions. Kim et al. [2] suggested a solution that makes use of attributes extracted from packet data to identify malicious activity. It was more likely that innocent packets wouldbe misidentified as malicious ones, and that malicious packets wouldbe mistaken for regular session traffic, if users took this strategy. To distinguish between malicious and benign sessions on a network, the suggested approach educates itself on the characteristics of malicious packets. When a GAN encounters a packet that it cannot properly categorize, it aborts the detection process until the next packet arrives. The proposed algorithm's accuracy in real-time network intrusion detection does not require the termination of a session or the delay time associated with collecting a predetermined number of packets, with a carefully crafted classification algorithm based on LSTM-DNN and a validation model using GAN.

Extreme difficulties are posed to intrusion detection systems by the variety of network threats. With a high false alarm rate (FAR), low recognition accuracy (ACC), and weak generalization capacity, traditional attack recognition systems typically utilize mining data correlations to find abnormalities. Hu et al. [3] presented a new intrusion detection approach using the adaptive synthetic sampling (ADASYN) algorithm and a refined convolution neural network to enhance the comprehensive capabilities of IDS and fortify network security. The author applied the ADASYN

technique to ensure a more uniform distribution of data points across the sample, which helps the model avoid overreacting to large or small samples. Second, the split convolution module (SPC-CNN) was the foundation of the enhanced CNN, which can expand the feature space and mitigate the effect of redundant inter-channel information on model training. When it comes to intrusion detection, the author turned to a model that combines AS-CNN, ADASYN, and SPC-CNN. Ultimately, the industry-standard NSL-KDD dataset was chosen to evaluate AS-CNN.

The Internet of Things (IoT) has exploded as a game-changing innovation ever since it was first conceived. IoT is the interconnection of computing devices and data to enable more process automation and centralization. The IoT is reshaping every aspect of business and culture. Further development of this technology raises the stakes for exploit detection and vulnerability awareness to safeguard vital infrastructure and operational procedures from being compromised and rendered inaccessible. Distributed Denial of Service (DDoS) assaults weremore widespread. Zeeshan et al. [4] proposed a Protocol Based Deep Intrusion Detection (PB-DID) architecture in this paper by comparing features from the UNSWNB15 and Bot-IoT datasets based on stream and Transmission Control Protocol (TCP). By resolving issues like unbalanced and over-fitting, the author was able to distinguish between normal, DoS, and DDoS traffic.

Supervisory control and data collection systems are used to manage and keep tabs on vital infrastructures including power plants, chemical refineries, and gas pipelines. For many years, researchers have been trying to figure out how to best identify intrusions in computer networks and other critical infrastructure, such as cyber-physical systems and industrial control systems. Viruses like seismic net, duqu, and flame against ICS assaults have wreaked havoc on nuclear facilities and other vital infrastructure in recent years. These stepped-up assaults have raised alarm bells about the safety of ICS systems in numerous nations. Dealing with unbalanced intrusion datasets, in which one class is represented by a smaller number of instances, was difficult when developing an intrusion detection framework. Therefore, Khan et al. [5] suggested an anomaly detection method for the ICS and explained a strategy to address this issue. The suggested method is based on a hybrid model that usesthe regular and predictable behavior of ground device communication in ICS environments. The author began by preparing the data to normalize and scale it. As a second step, anomaly identification was made more efficient by employing dimensionality reduction algorithms. Finally, the author used a modified version of the nearest-neighbor rule technique to ensure a more even distribution of values throughout the dataset. Finally, the Bloom filter is used to establish a signature database by monitoring the system for a predetermined time frame without any anomalies being recorded. To conclude, the author developed a hybrid technique for anomaly identification by combining this package contents-level detection with some other instance-based learner.

The complexity and variety of today's cyber attacks make it difficult to design an efficient system for intrusion detection in a multi-attack classification setting. Since cybercriminals can readily dodge the detection mechanisms set up in a network, Intrusion Detection Systems require highly efficient categorization techniques. In addition, it is difficult to effectively detect all types of attacks with a single classifier. Seth et al. [6] proposed an ensemble framework that was used to properly identify various types of attacks. To identify distinct kinds of attacks, the suggested method ranks the detection capabilities of different base classifiers. Calculating the rank matrix for various types of attacks requires using an algorithm's F1 score.

Parsamehr et al. [7] described an intrusion prevention system and location-aware prevention (IDLP) method that not only detects polluted packets and drops them, but also identifies the attacker's precise location to block them and avoid packet pollution in subsequent transmissions. The suggested

IDLP technique employs a homomorphic MAC scheme that relies on null space for both detection and localization. However, the suggested IDLP mechanism was effective since it does not need to be applied to all mobile devices in the first phase to safeguard the NC-enabled mobile small cells from resource depletion.

Mobile small cells equipped with NC are seen as an attractive technology for 5G networks due to theirpotential to reduce the financial and environmental burdens of these systems. Pollution attacks, in which intermediary nodes are used to tamper with packets en route, are a problem in an NC-enabled setting. Both spotting contaminated packets and tracking down fake users' physical locations are crucial for these networks. One of the most competitive systems for detecting pollution attacks and pinpointing the location of attackers in RLNC is Space-Mac. Here, Parsamehr et al. [8] evaluated Space-Mac against Parsamehret.alI DLP's method. Both algorithms have been incorporated into KODO, and the authors compared them concerning computational load, computational overhead, communication overhead, and decoding probability.

With an ever-increasing volume and variety of security breaches, ensuring the safety of a network is a top priority for its administrators and owners. Because of this dramatic increase, new forms of security must be devised. There are tools called Network Intrusion Detection Systems (NIDS) that monitor network traffic, analyze it for signs of attacks, and alert system administrators. In recent years, detection systems have benefited greatly from the incorporation of ML methods. However, ML approaches will have a detrimental effect on system performance because of the large number of features that must be examined to make sense of the complex data that is transferred over the networks. Improved NIDS performance was achieved by employing a feature selection strategy to pick the most important features from the input data. In this paper, Al-Zoubi et al. [9] advocated for anomalous network intrusion detection systems by proposing a wrapper strategy as a feature selection based on a Chaotic Crowd Search Algorithm (CCSA). The LITNET-2020 data set served as the basis for the experiments. The author believed this approach was the first selection technique based on swarm intelligence optimization to be used in this dataset to identify a unique subset of features for binary and multiclass classifications that simultaneously optimizes the performance for all classes.

The goal of any effective network intrusion detection system is to accurately identify malicious activity by identifying any deviations from the norm in system or network behavior. There are many obstacles to overcome while designing an adaptable and effective NIDS to counteract attacks that are sudden and unplanned. An intrusion detection system was developed using statistical, machine/deep learning-based methodologies. The majority of efforts are directed at boosting the model's precision while neglecting the false alarm rates. Shettar et al. [10] employed a model that combines Multilayer vision with chaotic neural networks to boost precision, accuracy, and reliability while simultaneously reducing false positives. Malware detection using MLP and false alarm suppression with chaos neural networks. The KDD Cup'99 dataset was used as a basis for the experiments.

Interference control, load sharing, and expanded capacity are just some of the issues that have prompted research to be conducted in various parts of the world. As a result of these obstacles, 5G is starting to look like a promising future option for meeting demand. Relays in device-to-device communication and small cell access points are two examples of new technologies that are developed to suit the rising demand. However, the adoption of these technologies has created new vulnerabilities in 5G wireless communication systems. Gupta et al. [11] explored the risks associated with 5G wireless networks and examined the impact of a bandwidth spoofing attack on the small cell access point within a 5G wireless network via the lens of game theory. An adaptive intrusion detection system based on a hidden Markov model was also proposed in this paper for use in 5G wireless communication networks to monitor small cell access points for unauthorized access.

Interference management, load sharing, and expanded capacity are just a few of the issues that have sparked international research efforts to address in the face of rising user numbers. As a result of these obstacles, 5G is starting to look like a promising future option for meeting demand. Relays in device-to-device communication and small cell access points are two examples of new technologies that have been developed to suit the rising demand. However, the introductions made had left 5G wireless communication networks vulnerable to attacks. The impact of a bandwidth spoofing assault on the small cell access point in a 5G wireless communication network was analyzed in this paper by Nguyen et al. [12] using game theory, with a focus on the security concerns of these networks. Furthermore, this study presented a hidden Markov model-based adaptive intrusion detection system for 5G wireless communication networks' small cell access points.

Due to the expansion of IoT, where a vast number of small, smart gadgets stream trillion bytes of data to the Internet, cyber security has become increasingly difficult. However, due to a lack of protection measures and hardware security support, these devices are susceptible to cyber-attacks. In addition, there is a severe lack of intrusion detection systems powered by deep learning in IoT gateways, which is a major vulnerability. Indeed, the computing requirements of deep learning models far outstrip the capabilities of such gateways. In this research, Singh et al. [13] presented Real Guard, a local gateway-based Network Intrusion Detection System (NIDS) based on a deep neural network (DNN) for protecting IoT devices. The proposed solution's strength is its real-time detection of multiple cyber attacks with low computational requirements. This is possible due to the use of a lightweight feature extraction mechanism and a powerful yet efficient attack detection model enabled by deep neural networks.

The Internet of Things is an emerging concept that describes the process by which the Internet is integrated into the various sectors of human civilization. However, due to their widespread nature, IoT networks are easy targets for cybercriminals. A Denial of Service (DoS) is a common form of attack in which the attacker sends an overwhelming amount of data to the network to overwhelm the system and prohibit any of the nodes from accessing the network's services. When it comes to safeguarding computer networks and other forms of IT infrastructure, intrusion detection systems are among the first lines of defense. However, due to constraints such as resource-constrained devices, the restricted memory and energy capacity of nodes, and particular protocol stacks, traditional intrusion detection algorithms need to be adapted and improved for use in the IoT. To detect an attacker trying to inject useless data into an IoT network, Jan et al. [14] created a lightweight attack detection approach using a supervised machine learning-based Support Vector Machine (SVM).

Harikrishnan et al. [15] proposed ChaosNet, a novel chaos-oriented artificial neural network design for classification tasks, inspired by the unpredictable nature of neuronal firing in the brain. Each neuron in ChaosNet is a layer of the network, and each layer is a Generalized Luroth Series (GLS) map, a 1D chaotic map whose features were demonstrated in other research to be particularly effective in compression, cryptography, and the computation of XOR and other logical operations. Using the topological transitiveness property of the chaotic GLS neurons, the author created a new learning algorithm for ChaosNet. The proposed method of learning achieves reliable high performance across a range of classification tasks, even when given only a small number of training samples from which to learn. ChaosNet achieves performance accuracies between 73.89% and 98.33% with as few as 7 training samples/class. The author offered a hypothetical development of a 2-layer ChaosNet for improving classification accuracy and showed that ChaosNet was robust to additive parameter noise.

3 Proposed Model

As NC can reduce packet transmission in the wireless multicast model, improve network capacity, and achieve robustness to packet losses with low energy consumption, it has the potential to deliver significant benefits to mobile networks. Despite these major benefits, NC small cells might be subjected to a wide variety of attacks due to the insecure nature of NC itself. An attacker can inject corrupted packets into an NC-enabled wireless network, making it impossible for the destination nodes to decode the packets correctly. In addition, the polluted packets can contaminate more packets on the network if they are permitted to travel through legitimate nodes. This not only leads to wasteful network resources and throughput utilization but also node energy consumption. Novel security methods against these intrusions in the NC-enabled cells are necessary for pinpointing the origin of polluted packets and identifying the precise location of malevolent users.

In this research, a chaotic sequence mapping based low overhead 1D Improved Logistic Map is used to secure peer-to-peer data transmission model using lightweight H-MAC with accurate intrusion detection. It is proposed to identify pollution assaults and denial of service attacks using a hash-based homomorphic message authentication code approach. The suggested model is focused on protecting NC-enabled networks from threats like denial-of-service and pollution attacks. In reality, the adversary is seeking to compromise the availability of the NC-enabled network and its nodes. The node or packet is discarded if the result of the verification is not zero in the proposed model that is used for accurate intrusion detection. The proposed model framework is shown in Fig. 4.

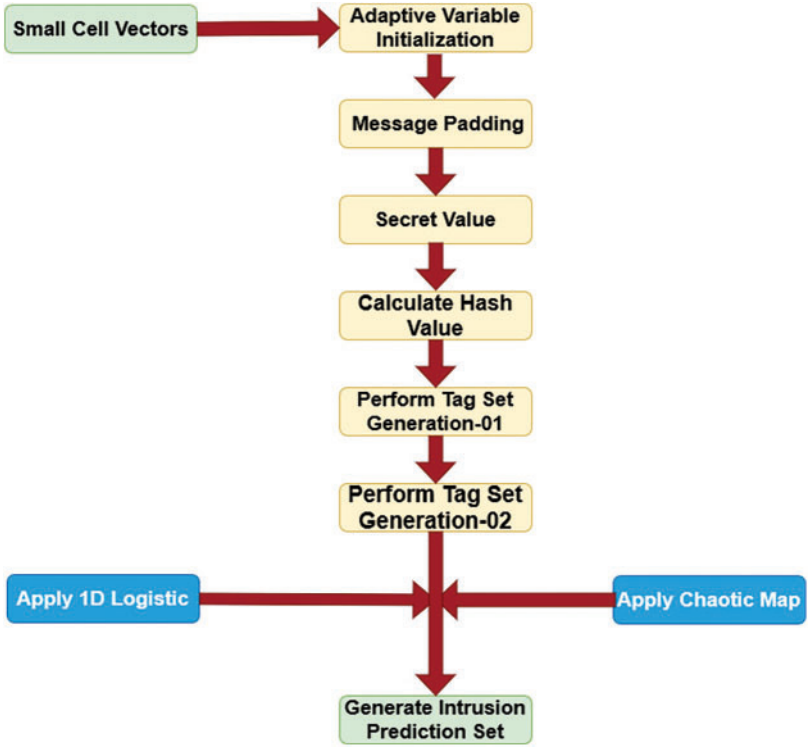

Figure 4: Proposed model framework

The SHA-256 enhanced model is used in this model for hash key generation. Patented in the field of cryptography, SHA-256 generates a value that is 256 bits in length. Safe Hash Algorithm, or

SHA, is an acronym for this. The Secure Hash Algorithm (SHA) is a cryptographic hashing algorithm that is a variant of MD5. Using bitwise operations, modularity additions, and compression functions, a hashing algorithm condenses the input data into a shorter, unintelligible version. One significant advantage of the suggested technology is that it may be implemented in realtime to secure 5G and future networks. To make reliable network intrusion detections, this study introduces an integrated intrusion detection system. The essential requirement that was foreseen is centered on the obligatory automatic real-time processing of massive amounts of data traffic that travel through the backbone of the telecoms provider's 5G network. Fortunately, the proposed algorithmic and architectural frameworks are in complete accordance with these requirements. In this scenario, a collection of intrusion detection systems denoted by

$SC = \{SC_1, SC_2..., SC_M\}$ are considered as small cells in the network that can be in varied range M.

Each node in the network is represented as SC which is capable of transmitting and receiving the data that are in range.

To improve upon the current network paradigm of store and forward, NC suggests implementing smarter routing that permits intermediary nodes to change data while in transit. The SC nodes in the network will consider the Hash-based homomorphic authentication codes. The source node will generate the vector set from the total packets to be transmitted. The vector sets are generated as batches where each batch has K packets set as

$$BV = \{PV_1, PV_2 \ldots., PV_K\} \tag{1}$$

The packet vector batches consider the sequential mapping of batches for data transmission that allows the batches to transmit to the neighbor nodes in the batch sequence.

$$DataTrans\,(BV\,[k]) = \sum_{p=1}^{K} \lim_{p \to PV_K} \left(\lambda + \frac{\omega}{p}\right)^{\tau} \tag{2}$$

Here p is the packets set considered from the batch set and λ is the sequence number of batch vector packets and ω is the next batch to be transmitted in the channel τ.

Computing and networking on a P2P basis is on the distributed application architecture, which divides up transmission loads and tasks across SC nodes. On top of the actual physical network structure, most P2P networks run some kind of virtual overlay network, whose nodes are only a subset of the total. During the P2P communication, to avoid intrusions in the network, HHMAC is used for authentication codes for accurate data transmission.

Since only those with possession of the private key can verify an HHMAC, the proposed research's key usage model is not publicly verifiable. HHMAC schemes allow anyone with access to the public evaluation key but not the secret key to validate a computation performed over already authenticated data by computing a short MAC. The secret key holder can thus validate the computation's outputs without needing access to the authenticated inputs themselves. To conduct the homomorphic computation across the authenticated data set, a hash-based homomorphic MAC scheme is simply used for accurate attack detection.

To obtain a MAC, HMAC uses a cryptographic hash function applied to the data being validated along with a secret shared key. It serves as a means of authentication and data integrity, just like every other MAC. All participants in a communication must perform data integrity checks. With HMACs, both the client and the server have access to a secret private key. Each request has its own special HMAC that is generated by the client. The client uses its private key to hash the data it requires from

the server before sending it. The message and the key are hashed using different algorithms, which provide an additional degree of security. In response to a request, the server creates its own HMAC. A fresh hash value for the received message can be generated by the recipient using the same HMAC algorithm and the same secret key to validate its authenticity. If the newly calculated hash value is the same as the previously calculated one, the message is legitimate and has not been tampered with. If the hash values are different, the message has likely been altered.

Lightweight MACs are composed of block cipher-centered, which means that at any given time, they process a block of communication to produce a block of cipher text. LightMAC is a lightweight MAC technique used for low-power devices that can send messages of varying lengths because its block size is not confined to 32 or 64 bits. This allows for a high message size per key. LightMAC's robustness and efficiency are unaffected by block size. The execution time for shorter messages is preferable to that of longer ones. When comparing M = 32 and M = 64, for instance, M = 8 has faster execution times. When M = 32, execution time is faster than when M = 64. For each M parameter, the execution speed is faster with a 128-bit key than with a 192-bit key.

To obtain a lightweight hash model the initial attributes are considered that calculate the hash value. The adaptive basic variable is considered as the result of XOR (⊕) between the random variable R1 and secret value SV.

$$\mathrm{AV} = \mathrm{R1} \oplus SV \tag{3}$$

The length of the message 'M' represented as 'ML' is dynamic and to satisfy the bit range, padding is performed based on adding the padding bits dynamically at the beginning and ending considering the MSB bit as '0' and the remaining P/2 bits as '1s' and LSB as '1' and remaining as '0 s'. It is recommended that the padded message be a divisor of len (ML) bits.

$$PBset = [0111\,||M||\,0001]$$

$$PB[M] = \sum_{p=1}^{K} \frac{PBset||R1}{len(ML)} \tag{4}$$

$$PadMsg[k] = \sum_{p,i=1}^{K} \frac{BV(p)\,||P(i) + PB}{len(BV)} \begin{cases} if\ (ML(p) == Th)\ return\ M \\ otherwise\ generate\ pading\ P \end{cases} \tag{5}$$

The secret value generation is performed by considering the basic vectors of length L, which is used for generating an HMAC. The input vectors are

$$VarK = \sum_{p=1}^{K} getPrime(p) \tag{6}$$

$$VarR = \sum_{p=1}^{K} getRand(p) \geq \max(VarK) \tag{7}$$

$$VarL = \sum_{p=1}^{K} \frac{get\ Min(p) > Var\ K\ \&\ get\ Min(p) < VarR}{len(Pad\ Msg)} \tag{8}$$

$$Pub_{Key} = \sum_{p=1}^{K} ((VarR \oplus \mathrm{VarK}) \oplus \mathrm{VarL}) \ll 2 \tag{9}$$

The secret value is generated based on the input parameters and the secret value is used one time for a batch and a new secret value is generated for every new batch 'V'.

$$SecVal = \sum_{BV=1}^{M} \frac{(\mathrm{VarK}*\mathrm{VarL}) \oplus \mathrm{VarR}}{\mathrm{VarR}\ \&VarL} * B^{\mathrm{R}} \tag{10}$$

Here B and R are threshold values considered for the secret key generation.

$$B = \sum_{p=1}^{K} getRand\,(p) < Pub_{Key} \tag{11}$$

$$R = \sum_{p=1}^{K} getPrime\,(p) > B \tag{12}$$

Based on the secret value, the hash model is calculated as

$$HV\,[BV\,(M)] = \sum_{p=1}^{K} \frac{VarL||VarK + VarR \oplus SecVal}{VarR \oplus \mathrm{VarL}} + SecVal^{R} \tag{13}$$

The Key Pair Set is generated as

$$KeyPair\,[M] = \sum_{p=1}^{K} \{(Pub_{Key}\,(p) \oplus \mathrm{HV\,(p)}) :\, SecVal(p) \oplus \mathrm{HV\,(p)}\} where\, p \in SCset \tag{14}$$

The valid node that owns a secret evaluation key can use a hash-based homomorphic message authenticator to compute over already-authenticated data, and the tag generated that results from this computation can be used to validate the authenticity of the computations itself.

$$Tagset\,[M] = \sum_{p=1}^{K} \{\overbrace{\left\{\mathrm{PV_1, PV_2, \ldots, PV_K}\right\}}^{BV_1}, \overbrace{\left\{\mathrm{PV_1, PV_2, \ldots, PV_K}\right\}}^{BV_2}, \ldots\ldots, \overbrace{\left\{\mathrm{PV_1, PV_2, \ldots, PV_K}\right\}}^{BV_M}\} \tag{15}$$

The key distribution to all the intermediate SC nodes is performed with a new key pair that is used for HMAC verification. The tags are generated for every data packet in a batch BV for message authentication to identify the attacks. The process of BV packet generation is performed as

$$Tag[M] = \frac{\sum_{p=1}^{K} PV\,(p) * (Pub_{Key}\,(p) \oplus \mathrm{HV\,(p)})}{(Pub_{Key}\,(p+1) \oplus \mathrm{HV\,(p+1)}} + HV\,[BV\,(p)] \tag{16}$$

$$\overline{Tag[M]} = \frac{\sum_{p,j=1}^{K} PV\,(p+1) * (Pub_{Key}\,(p+1+j) \oplus \mathrm{HV\,(p+1)})}{(Pub_{Key}\,(p+1-j) \oplus \mathrm{HV\,(p+1-j)}} + HV\,[BV\,(p+1)] \tag{17}$$

The 1D logistic map, a polynomial mapping of degree 2, is frequently used as a paradigmatic design of how complex; node chaotic behavior can emerge from very straightforward nonlinear dynamical methods. The basic logistic map of SC nodes with packet tags is represented as

$$L_{x+1} = G.Tag_x(1 - Tag_x) \tag{18}$$

where Tag_x is a positive adaptive attribute between zero and one that indicates the current state of the packet data.

The NC-enabled SC data transmission of packet vectors in batches BV are considered with the current state and the final tag updating is performed as

$$Tag\,[M] = \prod_{p=1}^{K} simm\left(\overline{Tag\,[p+1]}, Tag\,[p]\right) + mod(R * \min\left(Tagset^{Tag_x}\right) * HV\,[BV\,(p)]) \tag{19}$$

The 1D logistic map representation of the packets tags and SCs are defined with the differences in the tags identified during data transmission among the neighbor nodes. The process of the logistic map and the improved logistic map is performed as

$$LM = \prod_{p=1}^{K} \lim_{n\to\infty} \left(\max(Tag\,(p)\,, Tag\,(p+1)) + \frac{\lambda}{L_{x+1}} \right)^2 \tag{20}$$

$$ILM' = \prod_{p=1}^{K} mod(LM(G,p) - L_{x+1} - LM'(G,p+1)) * V \tag{21}$$

Here G is the packet parameter range of [0,4] and V is the range between $8 < V < 8 * N$.

A quadratic equation is considered that is used for performing enhanced logistic map in the network for intrusion detection schema. The equation is represented as

$$M = Zt - Zt^2$$

The factorized form is written as

$$M = Zt(1-t) \tag{22}$$

The non-linear quadratic model is represented based on the quadratic equation that needs to consider the limitation and growth factor of the packet tags as

$$T_{ILM(p+1)} = \sum_{p=1}^{K} \omega * T_{ILM}(1 - T_p) \tag{23}$$

Here ω is the growth factor and $(1 - T_p)$ is the batch vector packet limiting factor. The chaotic map is shown in Fig. 5.

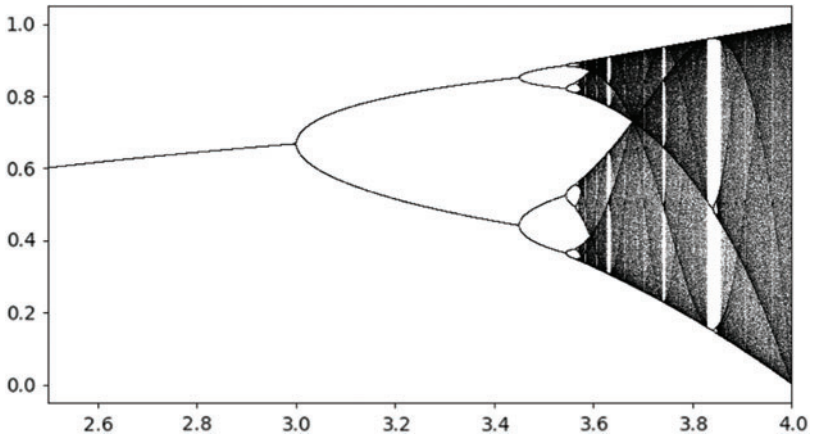

Figure 5: Chaotic map

Here u is the growth rate, and x is the population considered as the limiting factor for limiting the packets in the batch vector. Even in the short term, constraints on data packets affect the majority of individuals and these limits eventually strike. When the population size is small compared to U, there is little difference between the two patterns. For the subsequent population, however, the curves diverge as U approaches a large fraction of x, and the growth rate approaches zero as U approaches x. The logistic map growth factor and limiting factor are calculated as

Let $T_{ILM} \in [0,1]$ *and* ω *ranges from* $[0,8]$

$$\frac{dILM}{T_{ILM(p)}} = \sum_{p=1}^{K} C * \left(1 - \frac{U}{x}\right) + \max\,(Tag\,(p)) + \min(M) \tag{24}$$

Here C is the probability attribute, U is the growth factor and x is the limitation factor of packet vectors. The intrusion detection growth model can be updated as

$$X_r = \sum_{p=1}^{K} \frac{dILM(p)}{T_{ILM(p)}} + \frac{x * T_{ILM(p)}}{\tau + (U - \min(p)) * C^{-x}} \tag{25}$$

τ is the current population set in packet vectors.

A unique block cipher for protecting textual information was developed using a chaotic map. To achieve the diffusion and confusion qualities, packet text was first segmented into 16 bytes by 16 bytes blocks using padding. The chaotic map function process is performed by considering the Gaussian function to calculate the lightweight complexity in the chaotic map function that is performed as

$$S = mod\left(\frac{\max(ILM(p))}{1 + L_x * V}\right) \tag{26}$$

$$GauF(Tag[M]) = \sum_{p=1}^{K} \max_{8 \le V \le 8*N} ILM(p) * \exp\left(\frac{S(L_x)}{HV(p)}\right) \tag{27}$$

$$chaomap(LM[M]) = \prod_{p=1}^{K} \frac{2^{\mu} * \lambda}{2^{LM(p)*max(HV_p)}} + \lim_{p \to \infty}\left(\frac{\min(GauF(p))}{len(p)} log_2 \frac{\max(Tag(p))}{\pi * R}\right)^2 \tag{28}$$

S is the packet embedding vector; μ is the range of bit batches that areselected as multiples of 8.

The intrusion detection process is performed to identify the intrusions that occurred during the data transmission. The intrusion detection model accurately identifies the changes in the packets that occurwhen a malicious node triggers its actions in the network. The process of intrusion detection is performed as

$$\begin{aligned} Intd(Tag[M]) = &\sum_{p=1}^{K} \min(chaomap(ILM(p, p+1)) + \frac{\lambda}{L_{x+1}} \\ &+ count(PDR(S \longrightarrow R)) \begin{cases} 1 & if(diff(Tag(p), Tag(p+1)) > Th \\ 0 & Otherwise \end{cases} \end{aligned} \tag{29}$$

4 Results

The proposed model is implemented in the NS 2.35 simulator. The proposed model considers 40 nodes in the network as a simulation for intrusion detection. NS2 is an OTcl interpreter that takes an OTcl script as an input and outputs a trace file, making it a specialized event simulator for networking research. However, this assumption only holds for mobile nodes with high-rate and low-speed characteristics; for example, a node sending at 10 Kbps and moving at 10 m/s will move 12 m in the time it takes to receive a packet of 1500 B. The P2P model is simulated in the model that performs communication and intrusion detection in the network. The proposed model is tested on an Intel i5, 4 GB RAM, 3.0 GHz, Dual-Core system. A lightweight hash-based HMAC is proposed in this research for accurate intrusion detection. The malicious actor aims to compromise the operational status of the NC-enabled network and its nodes to degrade the performance. In this research, a chaotic sequence mapping based low overhead 1D Improved Logistic Map is used to secure peer-to-peer data transmission model using lightweight H-MAC (1D-LM-P2P-LHHMAC) is proposed with accurate intrusion detection. The proposed model is compared with the traditional intelligent

intrusion detection system that has been adapted to recognize or predict a cyber-attack using Elite Machine Learning algorithms (EML).

Attacks on devices, computer networks, or additional network assets that prevent authorized users from gaining access to those services and resources are known as DoS attacks. By overwhelming the target with traffic until it either crashes or stops responding, this goal is typically achieved. Companies and customers lose money and time due to downtime caused by DoS attacks, which can last anywhere from a few hours to several months. A DoS attack is one in which the target system or network is intentionally overloaded to the point where it cannot serve its intended purpose. DoS attacks succeed in this because they overwhelm the victim with traffic or deliver data that causes it to crash. Knowing that a DoS attack is happening is the first step in preventing more damage or halting the attack entirely. Gathering enough data about network traffic and then analyzing it to determine if the communication is friendly or malicious is necessary for detecting an attack. This can be done by hand or with automation software.

The source node will generate the vectors of packets for data transmission. The vectors are used to transfer batch wise to avoid loss. The Vector Set Generation Time Levels of the existing model and the proposed model are shown in Table 1 and Fig. 6.

Table 1: Vector set generation time levels

Nodes in the network	Models considered	
	1D-LM-P2P-LHHMAC model	EML model
50	7	12
100	9	14
150	11	16
200	13	18
250	15	20
300	17	22

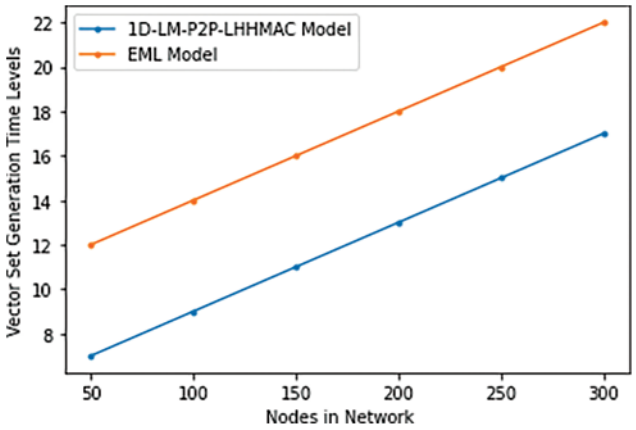

Figure 6: Vector set generation time levels

Sets are a special case of associative packet containers in which the value of each packet must be distinct. Once a node has been added to the set, its value cannot be changed, though removing it and re-adding it with a new value is an option. Table 2 and Fig. 7 represent the Vector Set Generation Accuracy Levels of the proposed and traditional models.

Table 2: Vector set generation accuracy levels

Nodes in the network	Models considered	
	1D-LM-P2P-LHHMAC model	EML model
50	88	80
100	90	81
150	90.8	82.7
200	92.6	85
250	95	86.8
300	97.5	89

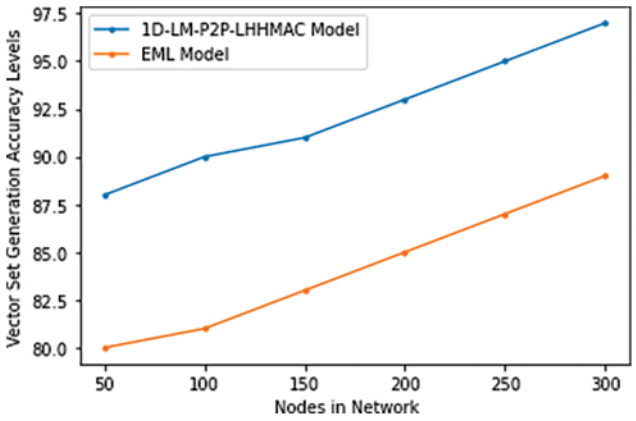

Figure 7: Vector set generation accuracy levels

Padding refers to a group of techniques in which information bits are added to the beginning and end of a message before it is transmitted. Data bits used to fill empty spaces in a field, packet, or frame are performed. Adding 1 bits, blank characters, or null characters as padding at the end of a data structure is a common practice for making it full to avoid header modifications. The Message Padding Accuracy Levels of the proposed and existing models are shown in Table 3 and Fig. 8.

Table 3: Message padding accuracy levels

Nodes in the network	Models considered	
	1D-LM-P2P-LHHMAC model	EML model
50	86.7	77.5
100	87.2	80.7

(Continued)

Table 3 (continued)

Nodes in the network	Models considered	
	1D-LM-P2P-LHHMAC model	EML model
150	89	81
200	91.3	82.6
250	93.2	84.7
300	95	86.3

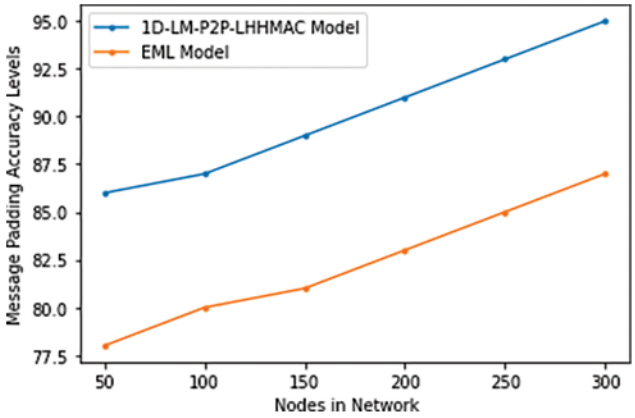

Figure 8: Message padding accuracy levels

The proposed model generates a key pair for data encoding for the transmission of packets. The key pairs are calculated using the hashing model and the lightweight operations increase the system performance levels. The key pairs keys can be used only once and the next keys will be used for other transactions. Table 4 and Fig. 9 depict the key pair generation time levels of the existing and proposed models.

Table 4: Key pair generation time levels

Nodes in the network	Models considered	
	1D-LM-P2P-LHHMAC model	EML model
50	9	14
100	11	16
150	12	17
200	14	18
250	15	20
300	16	22

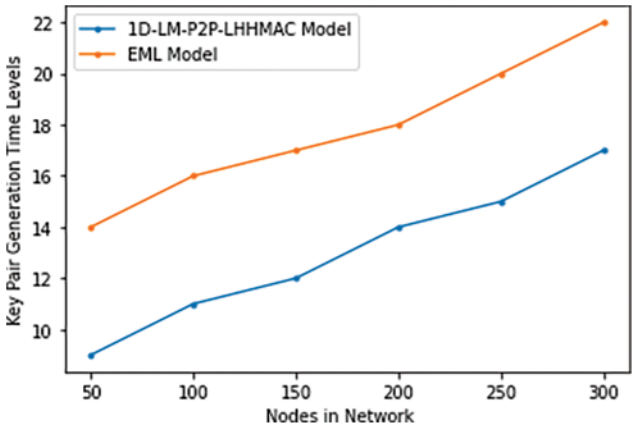

Figure 9: Key pair generation time levels

The network communication simulated in the NS2 simulator and the process of communication in the network is shown in Fig. 10. The intrusion detection model designed in this research identifies the intrusions accurately when each node is monitored in the network.

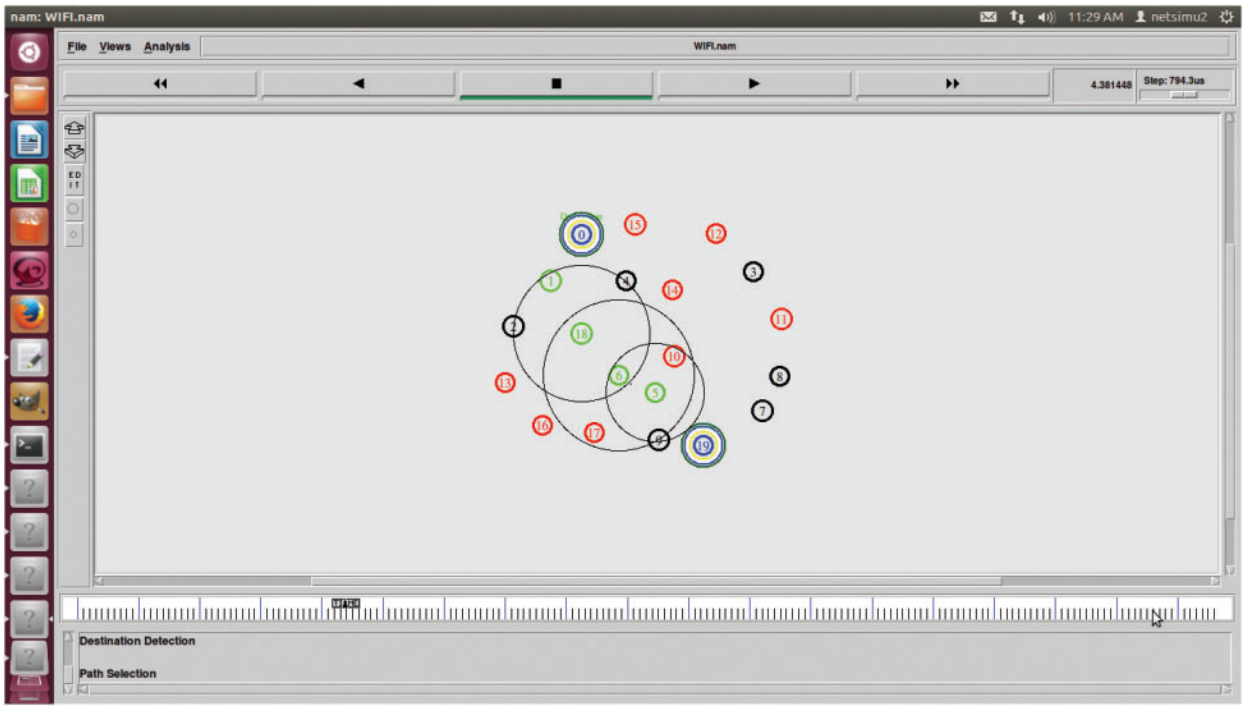

Figure 10: P2P communication in small cells

The logistic map is a discrete-time, one-dimensional map that is sophisticated for its apparent formal simplicity. A logistic map is used to project the population's value at one time step onto the value at the next time step, this mapping technique is known as the logistic map and as the value of μ shifts, the seeming simplicity of this equation belies a deeper complexity. Once surpasses a certain

threshold value μ, chaos ensues. The Improved Logistic Map Time Levels of the existing and proposed models are shown in Table 5 and Fig. 11.

Table 5: Improved logistic map time levels

Nodes in the network	Models considered	
	1D-LM-P2P-LHHMAC model	EML model
50	11	16
100	12.2	18
150	14.3	19
200	14.8	21
250	16	22
300	17.6	24

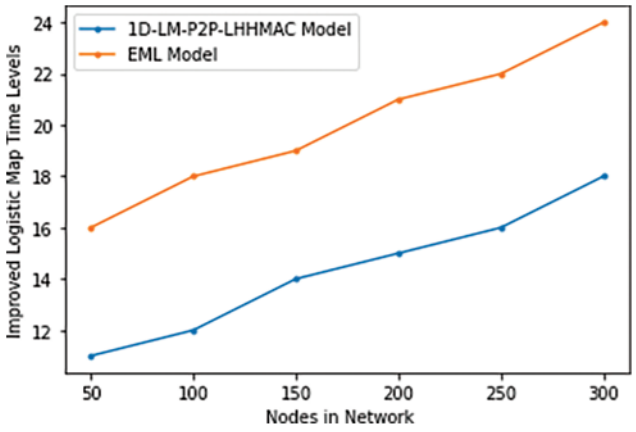

Figure 11: Improved logistic map time levels

Non-linear dynamic systems, of which chaotic systems are examples, exhibit erratic behavior. Pseudo-random number generators and multimedia encryption both make extensive use of chaotic maps. The goal of chaos theory is to provide a quantitative and qualitative framework for studying the behavior of dynamic systems for which only holistic, continuous data linkages can provide adequate explanation and prediction. A discrete chaos map's main benefit is its straightforward mathematical model, which can be easily implemented and computed. When given an initial seed value, chaotic maps are mathematical operations that produce a seemingly random pattern. Since data science is becoming increasingly important, it is important to keep it secure. Table 6 and Fig. 12 represent the Applying Chaotic Map Accuracy Levels of the proposed and existing models.

Table 6: Applying chaotic map accuracy levels

Nodes in the network	Models considered	
	1D-LM-P2P-LHHMAC model	EML model
50	88.6	81
100	91.8	81.5
150	93	83
200	94.8	86.7
250	96.4	88
300	98	89.5

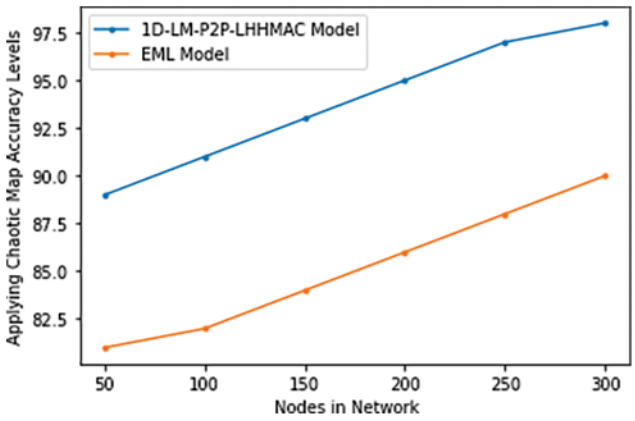

Figure 12: Applying chaotic map accuracy levels

The chaos growth vector limiting factor is set that is used for the accurate detection of attacks. The Chaos Growth Vector Limiting Factor Calculation Accuracy Levels of the proposed and existing models are shown in Table 7 and Fig. 13.

Table 7: Chaos growth vector limiting factor calculation accuracy levels

Nodes in the network	Models considered	
	1D-LM-P2P-LHHMAC model	EML model
50	97.2	93
100	97.5	93.4
150	97.8	93.6
200	98	93.8
250	98.1	94.6
300	98.2	95

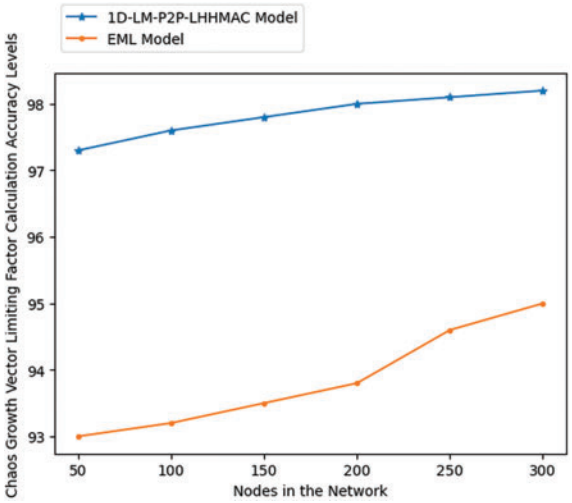

Figure 13: Chaos growth vector limiting factor calculation accuracy levels

The normal distribution known as the Gaussian distribution, is symmetrical around the mean probability distribution showing that data points closer to the mean occur more frequently than data points further away from the mean. The Gaussian function attribute calculation is used in designing the chaotic curve for the detection of the probability of attacks in the network. The Gaussian Function Calculation Time Levels of the proposed and existing models are shown in Table 8 and Fig. 14.

Table 8: Gaussian function calculation time levels

Nodes in the network	Models considered	
	1D-LM-P2P-LHHMAC model	EML model
50	12	21
100	13	21.3
150	13.6	22
200	13.8	22.2
250	14	22.4
300	14.2	22.6

The intrusion detection mode in peer-to-peer communication is triggered based on the logistic mapping and chaotic curve that detects the minute changes in the range of packet vector bits that improve the network performance. By either directly notifying security administrators of known or suspected threats, or by delivering alerts to a centralized security tool like a security information and event management system, the proposed IDS can help speed up and automate the process of detecting and responding to cyber threats. The proposed IDS is designed to keep monitoring the network's resources and alert users about any suspicious activity. Attackers are always coming up with new ways to get around defenses and have an impact. In some cases, hackers try to steal login credentials so they can access protected systems and information. When malicious activity is detected, it may be dealt with quickly and efficiently with Network IDS.The primary benefit of an IDS is alerting the appropriate person when an attack occurs. A network intrusion detection system also monitors

information passing between the system and the network to ensure that no malicious activity is taking place. The Intrusion Detection Accuracy Levels of the existing and proposed models are shown in Table 9 and Fig. 15.

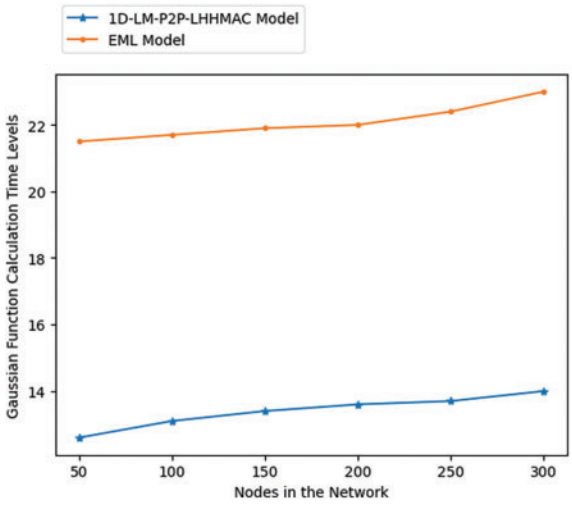

Figure 14: Gaussian function calculation time levels

Table 9: Intrusion detection accuracy level

Nodes in the network	Models considered	
	1D-LM-P2P-LHHMAC model	EML model
50	85	77
100	86.5	79
150	88	81
200	91	82.5
250	92.5	83
300	95	85.5

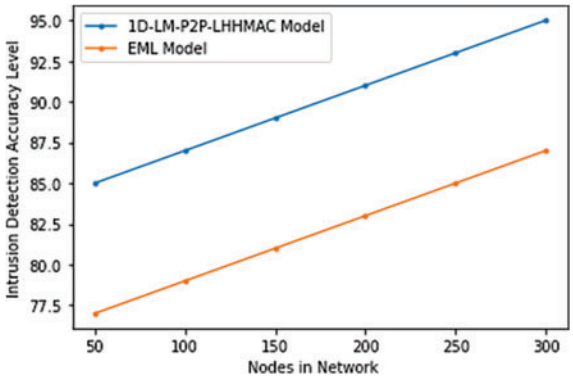

Figure 15: Intrusion detection accuracy level

The testing loss shows how effectively the model generalizes to new data, whereas the training loss shows how well it generalizes to the original training data. Training data is used to train a model, whereas test data is used to verify the model's accuracy on novel data. However, misunderstandings might arise due to the parallels and distinctions between the two. Users can reduce the impact of data discrepancies and gain a deeper understanding of the model's properties by utilizing the same data for training and testing. The proposed model is put to the test once it has been processed using the training set by making intrusion predictions on the test set. Table 9 and Fig. 15 show the Training and Testing Loss Levels.

5 Conclusion

The importance of user data security is growing as more advanced internet technologies become widely available. Because of the rapid spread of network technology, security measures are becoming increasingly important everywhere. The growth and extensive consumer use of the WSN depend on secure, private networks that protect user data. IDS helps spot Intruder attempts and other types of network-based attacks. This research utilizes a chaotic sequence mapping based low overhead 1D Improved Logistic Map to secure peer-to-peer data transmission model using lightweight H-MAC with accurate intrusion detection. The proposed method is only relevant to tagged nodes when there is no attack in the network. When an attack is absent, this approach can avoid the time spent verifying at each intermediate node. The proposed method employs a hash-based homomorphic MAC technique for detecting attacks in the network, which provides a serious security issue in NC-enabled networks. The proposed system discards discovered faulty packets and stops the attacker from spreading the corrupted packet in the network, thus protecting the NC-enabled mobile small cells from the waste of network resources and energy. Nevertheless, this is not enough, as attackers may continue to waste throughput by causing malicious actions in the network in succeeding rounds. In the future, there is potential to incorporate optimization models and adopt hybrid crypto mechanisms for key generation, management, and securing data within the network.

Acknowledgement: None.

Funding Statement: Self-funded; no external funding is involved in this study.

Author Contributions: Study conception and design: Chanumolu Kiran Kumar; data collection: Chanumolu Kiran Kumar; analysis and interpretation of results: Nandhakumar Ramachandran; draft manuscript preparation: Chanumolu Kiran Kumar. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this paper can be obtained from the corresponding author upon request.

Conflicts of Interest: The authors declare they have no conflicts of interest to report regarding the present study.

References

[1] H. Siddharthan, T. Deepa and P. Chandhar, "SENMQTT-SET: An intelligent intrusion detection in IoT-MQTT networks using ensemble multi cascade features," *IEEE Access*, vol. 10, pp. 33095–33110, 2022. https://doi.org/10.1109/ACCESS.2022.3161566

[2] T. Kim and W. Pak, "Early detection of network intrusions using a GAN-based one-class classifier," *IEEE Access*, vol. 10, pp. 119357–119367, 2022. https://doi.org/10.1109/ACCESS.2022.3221400
[3] Z. Hu, L. Wang, L. Qi, Y. Li and W. Yang, "A novel wireless network intrusion detection method based on adaptive synthetic sampling and an improved convolutional neural network," *IEEE Access*, vol. 8, pp. 195741–195751, 2020. https://doi.org/10.1109/ACCESS.2020.3034015
[4] M. Zeeshan, Q. Riaz, M. A. Bilal, M. K. Shahzad, J. Hajira *et al.,* "Protocol-based deep intrusion detection for DoS and DDoS attacks using UNSW-NB15 and Bot-IoT data-sets," *IEEE Access*, vol. 10, pp. 2269–2283, 2022. https://doi.org/10.1109/ACCESS.2021.3137201
[5] I. A. Khan, D. Pi, Z. U. Khan, Y. Hussain and A. Nawaz, "HML-IDS: A hybrid-multilevel anomaly prediction approach for intrusion detection in SCADA systems," *IEEE Access*, vol. 7, pp. 89507–89521, 2019. https://doi.org/10.1109/ACCESS.2019.2925838
[6] S. Seth, K. K. Chahal and G. Singh, "A novel ensemble framework for an intelligent intrusion detection system," *IEEE Access*, vol. 9, pp. 138451–138467, 2021. https://doi.org/10.1109/ACCESS.2021.3116219
[7] R. Parsamehr, G. Mantas, J. Rodriguez and J. F. Martínez-Ortega, "IDLP: An efficient intrusion detection and location-aware prevention mechanism for network coding-enabled mobile small cells," *IEEE Access*, vol. 8, pp. 43863–43875, 2020. https://doi.org/10.1109/ACCESS.2020.2977428
[8] R. Parsamehr, G. Mantas, J. Rodriguez and J. F. Martínez-Ortega, "On the performance analysis of IDLP and spacemac for network coding-enabled mobile small cells," *IEEE Communications Letters*, vol. 25, no. 2, pp. 407–411, 2021. https://doi.org/10.1109/LCOMM.2020.3027972
[9] H. Al-Zoubi and S. Altaamneh, "A feature selection technique for network intrusion detection based on the chaotic crow search algorithm," in *2022 Int. Conf. on Intelligent Data Science Technologies and Applications (IDSTA)*, San Antonio, TX, USA, pp. 54–60, 2022. https://doi.org/10.1109/idsta55301.2022.9923108
[10] P. Shettar, A. V. Kachavimath, M. M. Mulla, N. D. G and G. Hanchinmani, "Intrusion detection system using MLP and chaotic neural networks," in *2021 Int. Conf. on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, pp. 1–4, 2021. https://doi.org/10.1109/ICCCI50826.2021.9457024
[11] A. Gupta, R. K. Jha and S. Jain, "Attack modeling and intrusion detection system for 5G wireless communication network," *International Journal of Communication Systems*, vol. 3, no. 10, pp. 1–14, 2017.
[12] X. H. Nguyen, X. D. Nguyen, H. H. Huynh and K. H. Le, "Realguard: A lightweight network intrusion detection system for IoT gateways," *Sensors*, vol. 22, no. 2, pp. 432, 2022. https://doi.org/10.3390/s22020432
[13] P. Singh, A. Kaur and S. K. Pal, "A novel chaotic flower pollination-based intrusion detection framework," *Soft Computing*, vol. 24, pp. 16249–16267, 2020. https://doi.org/10.1007/s00500-020-04937-1
[14] S. U. Jan, S. Ahmed, V. Shakhov and I. Koo, "Toward a lightweight intrusion detection system for the Internet of Things," *IEEE Access*, vol. 7, pp. 42450–42471, 2019. https://doi.org/10.1109/ACCESS.2019.2907965
[15] N. B. Harikrishnan, K. Aditi, S. Snehanshu and N. Nithin, "ChaosNet: A chaos based artificial neural network architecture for classification," *Chaos*, vol. 29, pp. 113125, 2019. https://doi.org/10.1063/1.5120831
[16] D. Zheng, Z. Hong, N. Wang and P. Chen, "An improved LDA-based ELM classification for intrusion detection algorithm in IoT application," *Sensors*, vol. 20, no. 6, pp. 1706, 2020. https://doi.org/10.3390/s20061706
[17] C. Iwendi, S. Khan, J. H. Anajemba, M. Mittal, M. Alenezi *et al.,* "The use of ensemble models for multiple class and binary class classification for improving intrusion detection systems," *Sensors*, vol. 20, no. 9, pp. 2559, 2020. https://doi.org/10.3390/s20092559
[18] N. Marir, H. Wang, G. Feng, B. Li and M. Jia, "Distributed abnormal behavior detection approach based on deep belief network and ensemble SVM using spark," *IEEE Access*, vol. 6, pp. 59657–59671, 2018. https://doi.org/10.1109/access.2018.2875045
[19] K. Wu, Z. Chen and W. Li, "A novel intrusion detection model for a massive network using convolutional neural networks," *IEEE Access*, vol. 6, pp. 50850–50859, 2018. https://doi.org/10.1109/access.2018.2868993
[20] P. Nancy, S. Muthurajkumar, S. Ganapathy, S. V. N. S. Kumar, M. Selvi *et al.,* "Intrusion detection using dynamic feature selection and fuzzy temporal decision tree classification for wireless sensor networks," *IET Communications*, vol. 14, no. 5, pp. 888–895, 2020. https://doi.org/10.1049/iet-com.2019.0172

[21] T. Su, H. Sun, J. Zhu, S. Wang and Y. Li, "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset," *IEEE Access*, vol. 8, pp. 29575–29585, 2020. https://doi.org/10.1109/access.2020.2972627

[22] P. Wei, Y. Li, Z. Zhang, T. Hu, Z. Li *et al.,* "An optimization method for intrusion detection classification model based on deep belief network," *IEEE Access*, vol. 7, pp. 87593–87605, 2019. https://doi.org/10.1109/access.2019.2925828

[23] M. Gao, L. Ma, H. Liu, Z. Zhang, Z. Ning *et al.,* "Malicious network traffic detection based on deep neural networks and association analysis," *Sensors*, vol. 20, no. 5, pp. 1452, 2020. https://doi.org/10.3390/s20051452

[24] H. Yang and F. Wang, "Wireless network intrusion detection based on improved convolutional neural network," *IEEE Access*, vol. 7, pp. 64366–64374, 2019. https://doi.org/10.1109/access.2019.2917299

[25] T. Yi, X. Chen, Y. Zhu, W. Ge and Z. Han, "Review on the application of deep learning in network attack detection," *Journal of Network and Computer Applications*, vol. 212, pp. 103580, 2023. https://doi.org/10.1016/j.jnca.2022.103580

[26] M. Gopinath and S. C. A. Sethuraman, "Comprehensive survey on deep learning based malware detection techniques," *Computer Science Review*, vol. 47, pp. 100529, 2023. https://doi.org/10.1016/j.cosrev.2022.100529

[27] M. A. Ferrag, L. Maglaras, S. Moschoyiannis and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches datasets and comparative study," *Journal of Information Security and Applications*, vol. 50, pp. 102419, 2020. https://doi.org/10.1016/j.jisa.2019.102419