Annals of Telecommunications
https://doi.org/10.1007/s12243-026-01203-7

Learning at the nano-scale: how to dynamically protect data in nano-network transmissions

Alessandra Rizzardi[1] · Sabrina Sicari[1] · Luigi Alfredo Grieco[1] · Giuseppe Piro[2] · Jesus F. Cevallos-Moreno[1] · Alberto Coen-Porisini[1]

Received: 9 February 2026 / Accepted: 23 July 2026

Abstract

Data transmission at the nano-scale faces constraints that are notably different from those of conventional wireless systems, especially in terms of communication reliability, resource availability, and security. In the Internet of Nano-Things (IoNT) environments, protecting sensitive information while preserving acceptable network performance is still an open challenge, particularly when cryptographic mechanisms introduce additional processing and communication overhead. To address such an issue, this paper proposes an IoNT framework that integrates a DNA-based bio-molecular cryptographic scheme with a machine learning (ML) module aimed at dynamically predicting suitable security configurations under changing network conditions. The proposed approach is evaluated in a healthcare scenario based on a simulated artery environment, considering different routing/MAC combinations and varying numbers of nano-devices. Results show that the integration of ML enables improved robustness under congestion conditions, reducing packet loss by approximately 33–37% across the evaluated scenarios. Such an improvement comes at the cost of an additional latency overhead, ranging from about 18% in dense deployments to approximately 55% in smaller-scale scenarios. At the same time, the DNA-based scheme provides enhanced protection at the expense of a configuration-dependent delay increase, which becomes more evident as the number of nano-devices grows. Overall, the findings highlight the feasibility of combining adaptive ML-driven control with bio-inspired cryptography to balance security and performance in dynamic IoNT systems.

Keywords Internet of Nano-Things · Nano-networks · Cryptography · Machine learning

✉ Alessandra Rizzardi
alessandra.rizzardi@uninsubria.it

Sabrina Sicari
sabrina.sicari@uninsubria.it

Luigi Alfredo Grieco
alfredo.grieco@poliba.it

Giuseppe Piro
giuseppe.piro@poliba.it

Jesus F. Cevallos-Moreno
jf.cevallosmoreno@uninsubria.it

Alberto Coen-Porisini
alberto.coenporisini@uninsubria.it

[1] Dip. di Scienze Teoriche e Applicate, Università degli Studi dell'Insubria, Via O.Rossi 9, 21100 Varese, Italy

[2] Dip. di Ingegneria Elettrica e dell'Informazione, Politecnico di Bari, Via Orabona 4, 70125 Bari, Italy

1 Introduction

Internet of Nano-Things (IoNT) is an emerging paradigm that envisions the integration of the nano-technologies into the Internet of Things (IoT) networks [1]. IoNT finds application in multiple domains, including, among others, bio-medicine and biomedical engineering, healthcare, industry, agro-food, and military/defense scenarios [2]. Inside a nano-network, *nano-devices* are usually able to perform both sensing and actuation tasks. Some of them may also act as *nano-routers*, enabling them to forward data to other devices, such as an access point, a sink node, a smartphone, and so on. A nano-device is expected to have very limited capabilities; thus, it can only execute simple sensing, computing, actuation, and information storage tasks. To overcome such limits, a set of nano-devices and nano-routers usually cooperate with each other within the same nano-network.

Published online: 05 August 2026

The work presented in this paper starts from the adoption of the IoNT framework introduced in [3]. Such a system integrates, within the same nano-network, electromagnetic transmissions, molecular exchanges, and wi-fi communications. Such heterogeneity raises security and privacy issues, mainly in the presence of sensitive information [4]. An IoNT system may be vulnerable to data breaches (e.g., eavesdropping of sensitive information, altering of transmissions, privacy leakage) and to attacks against network resources (e.g., traffic alteration, denial of service, man-in-the-middle). To cope with such threats at the nano-scale level, the IoNT framework has been enriched with a bio-molecular end-to-end cryptography mechanism. It is based on the properties of DNA and proteins [5], protects the interaction between nano-devices and nano-routers, supports data integrity and confidentiality requirements in a lightweight manner, as reported in [6], and addresses resource constraints affecting nano-devices. However, optimizing the nano-network's configuration parameters is crucial to ensuring the long-term performance of the IoNT system. A continuous monitoring activity would enable proper tuning of the network's settings based on the current system status, for example, in the presence of congestion or when anomalies are detected. Hence, this paper proposes integrating a prediction mechanism, based on machine learning (ML) techniques, into the existing IoNT framework to monitor the nano-network's behavior and performance, thereby allowing us to infer the robustness and efficiency of the generated ciphertext during system running.

The conducted analysis and the outcomes of the ML module are evaluated through a case study in the healthcare domain. Note that the nano-network allows different settings to be activated for ciphertext generation and routing algorithms. It is worth noting that this work significantly extends our previous study [6] presented at the 13th Wireless Days Conference, held at the Fluminense Federal University (UFF), in Niterói (Rio de Janeiro) on 1st–3rd December 2025. In fact, while [6] focused primarily on the integration and initial assessment of a DNA-based cryptographic mechanism for protecting nano-network transmissions, the present proposal introduces an ML-based predictive layer for adaptive parameter selection, expands the evaluation to include robustness-oriented metrics, and provides a broader analysis of the trade-off between security and network performance.

Despite recent advances in IoNT architectures, an important open issue is how to dynamically tune security-related parameters in a way that preserves both communication efficiency and cryptographic robustness under changing network conditions. This challenge is particularly relevant in nano-networks, where devices are severely constrained, and the introduction of security mechanisms may significantly affect delay, scalability, and reliability. In order to cope with this challenge, the paper aims to investigate whether a ML-driven control strategy can support the adaptive use of a DNA-based cryptographic mechanism in IoNT environments, so as to balance security and network performance in a dynamic and resource-constrained setting. Hence, unlike previous studies, our goal goes beyond the mere combination of a DNA-based cryptographic mechanism and ML. In fact, we introduce an adaptive security-control framework in which ML is used to predict the joint impact of cryptographic and networking parameters on both communication performance and ciphertext robustness, thereby enabling dynamic security configuration in resource-constrained IoNT environments. Such a behavior transforms the role of ML from a monitoring or detection tool into a decision-support mechanism for runtime security adaptation.

The main contributions of this work can be summarized as follows:

- A secure IoNT framework is integrated with a predictive ML module able to estimate the impact of security-related parameters on both performance and ciphertext robustness.
- The joint effect of DNA strand length and molecular control interval is analyzed on delay and randomness-related metrics.
- The proposed framework is evaluated under different routing and MAC configurations in a healthcare-oriented IoNT scenario.

The paper is structured as follows. Section 2 discusses the state of the art at the intersection of nano-technology applications, security, and ML. Section 3 provides details on the IoNT framework, which is integrated with a bio-molecular cryptographic algorithm, followed by Sect. 4, which is about the adopted ML technique. Section 5 presents the performance evaluation and discusses the trade-off between increased data protection and the communication overhead introduced by the combined cryptographic and predictive mechanisms; also, it analyzes the current limitations of the approach and outlines the directions required for broader validation and deployment. Section 6 ends the paper, providing hints for future research directions.

2 Related work

The intersection between security mechanisms and ML techniques in nano-scale communication systems is still an emerging research area. On the one hand, DNA cryptography at the nano-scale still remains an open research topic [7–10]. Also, the interconnection and interoperability of the nano-devices with existing communication networks and paradigms require the design and development of new IoNT infrastructures and standards. On the other hand, integrating

ML techniques inside the nano-network is a complex-prone task, mainly due to the resource constraints that affect this kind of environment. In the following, we provide an analysis of the state of the art in both cryptography and ML applied to nano-technology. Existing contributions can be categorized into two main directions: (i) DNA-based and bio-inspired cryptographic approaches and (ii) ML-driven solutions for nano-network optimization and security.

2.1 Cryptography at the nano-scale

The authors of [11] coined the term biochemical cryptography to emphasize the need for new security and cryptographic solutions in the field of nano-communications. In fact, actual approaches might not be applicable due to antenna size and channel limitations, as well as limited available memory and processing capabilities of the so-called nano-machines. Instead, in [12], the authors present an architecture for trusted remote sensing in *Public Physical Unclonable Function (PPUF)*, based on nano-technology. The proposed security protocol consists of authentication and time-stamping in order to counteract statistical guessing attacks. DNA origami is proposed in [13] to secure communications by means of some bio-molecular cryptography principles, based on protein binding-based steganography and self-assembled braille-like patterns.

The authors of [14] propose a DNA-based cryptographic method, which consists of a symmetric algorithm based on pseudo-DNA cryptography and molecular biology. Splicing and padding techniques are combined with complementary rules to make the algorithm secure by means of an additional layer of security compared to conventional cryptographic techniques. The work presented in [15] proposes to improve the key strength by combining the genetic algorithm with the Diffie-Hellman key exchange algorithm, while a C++ algorithm for DNA cryptography is envisioned in [16], even if their feasibility is neither evaluated nor tested. However, important information is provided concerning the speed and storage of DNA computing. In particular, the authors state that a conventional computer can compute at the rate of approximately 100 million instructions per second (MIPS), but experimentally, it is found that DNA strand combinations are generated by combining DNA strands on a computer at the rate of 109 MIPS or 100 times faster than the fastest computer. Instead, media storage requires 10–12 nm^3 to store 1 bit, while DNA needs only 1 bit per cubic nano-meter, so very large amounts of data can be stored in a compact volume.

The work in [17] proposes a modified lightweight DNA-based encryption method, which employs the randomization nature of the DNA sequences to generate the encryption key; such an encryption key will be exploited to make the encryption process simpler and faster to accommodate the computations of IoT devices. Experiments have been conducted on image encryption processing. A similar evaluation is described in [18], where a lightweight DNA-based encryption method and the elliptic curve encryption (ECC) are combined to secure IoT communications. Instead, in [19], secure data storage is provided in the form of DNA sequences; more in detail, a DNA-based Cryptographic Security Framework incorporates a robust key agreement protocol and encryption algorithm, which have been integrated into the CloudSim simulator. The claimed target domain is health cloud data, even if experiments do not present an e-health scenario.

Finally, the work presented in [6] is specifically targeted to IoNT. Unlike existing approaches, it enables the verification of DNA cryptography efficacy in a running nano-network, considering relevant network conditions, such as the routing protocol, the number of devices, and the packet rate. The authors of [20] focus on drug delivery in the human body, in a scenario where the drug molecule is protected with the help of protein cryptography and a genetic signature. MATLAB is used for simulations, while the work in [6] represents a starting point for the design and development of complex and effective cryptographic solutions by means of ad-hoc simulators, to provide security and privacy functionalities in heterogeneous IoNT contexts.

What emerges is that bio-molecular cryptography represents a promising alternative to traditional approaches in nano-scale environments, where conventional encryption schemes may be unsuitable due to strict resource constraints. Several works have explored DNA-based techniques to achieve secure data encoding by exploiting the combinatorial properties of biological sequences. For instance, pseudo-DNA cryptographic schemes and hybrid methods combining DNA encoding with conventional techniques have been proposed to enhance key generation and confidentiality. More recent studies have investigated lightweight DNA-based encryption tailored to IoT environments, focusing on reducing computational complexity while preserving security guarantees. Despite such advancements, many solutions are limited to conceptual designs or application-specific implementations (e.g., image encryption or cloud storage security), often without fully considering the interaction with dynamic network conditions. Moreover, scalability and integration with network protocols are rarely addressed in a unified framework.

2.2 Machine learning for nano-scale cybersecurity

The convergence of ML with the IoNT and the Internet of Bio-Nano-Things (IoBNT) represents a critical research frontier dedicated to managing the inherent complexities of data processing and network protection at the nano-scale [21]. The current literature emphasizes the role of ML in opti-

mizing communication protocols and enhancing biomedical applications through robust security frameworks [22].

For instance, the work presented in [23] introduces a parameter profiling framework utilizing decision trees for anomaly detection, while later proposing deep learning architectures to secure bioluminescent communication systems [24]. Similarly, the authors of [25] integrate deep learning with meta-heuristic optimisation to identify anomalous activities in IoBNT environments, and the authors of [26] provide a comparative analysis of classical supervised models (including random forest and SVM) for traffic classification in electromagnetic nano-networks.

Beyond intrusion detection, research has also targeted physical-layer challenges; the work in [27] explores 5G-enabled ML paradigms for route optimisation, whereas the work proposed in [28] benchmarks ML techniques to profile nano-scale signal attenuation and manipulation. Despite these advancements, focusing on threat detection and signal profiling, our work envisions using ML to rapidly adjust cybersecurity parameters in response to the volatile conditions of a dynamic IoNT environment.

Hence, ML techniques have been increasingly applied to nano-network environments, primarily for anomaly detection, signal classification, and performance optimization. Decision-tree-based models, deep learning architectures, and hybrid optimization techniques have been proposed to enhance traffic analysis, intrusion detection, or communication reliability. However, such approaches typically focus on detecting abnormal behaviors or improving physical-layer performance, rather than actively controlling the configuration of security mechanisms.

2.3 Research gap and contribution

Overall, the literature shows that existing solutions mainly address either DNA-inspired cryptographic protection or ML-based monitoring/classification tasks in nano-scale and bio-nano communication settings. However, only limited attention has been devoted to the joint use of adaptive ML mechanisms to support the dynamic configuration of security parameters in an operational IoNT framework. Most ML-based solutions are designed for classification or prediction tasks, without being directly integrated into adaptive control loops capable of tuning cryptographic parameters in real time, as proposed in this paper. Also, the envisioned solution extends the role of ML from passive monitoring to active control of security parameters, thereby enabling dynamic adjustment of the system configuration according to network conditions.

In this perspective, the novelty of the proposed work lies in the introduction of an adaptive control loop that bridges the gap between bio-inspired cryptography and network-aware intelligence. Existing DNA-based approaches generally employ static cryptographic configurations, while ML-based solutions in IoNT and IoBNT are primarily designed for anomaly detection, traffic classification, or parameter profiling. In contrast, the proposed framework adopts ML predictions to estimate the effect of security-related parameters on both network behavior and cryptographic robustness before configuration changes are applied.

To better position the proposed approach, we make a comparison with related work into two complementary perspectives. Table 1 summarizes the context, cryptographic techniques, and role of ML in existing approaches, while Table 2 highlights their main limitations and the key differences with respect to our proposal. As shown in Tables 1 and 2, existing approaches either focus on bio-inspired cryptography without adaptive capabilities or exploit ML mainly for detection purposes. Our approach integrates both aspects into a unified and adaptive framework.

3 IoNT security-aware framework

This section describes the conceived IoNT framework along with its integration with DNA cryptography. The reference architecture is sketched in Fig. 1 and includes the following components:

- A *smart device*, which is not affected by power constraints or resource limitations. It is responsible for

Table 1 Comparison with related works in terms of features

Ref.	Context	Cryptographic method	ML role
[17]	IoT (image encryption)	DNA-based lightweight	None
[18]	IoT (communication systems)	DNA + ECC	None
[19]	Cloud/Healthcare (simulation)	DNA-based framework	None
[6]	IoNT (healthcare)	DNA-based crypto	None
[23]	IoBNT (anomaly detection)	None	Decision trees
[24]	IoBNT (signal security)	None	Deep learning
[26]	Nano-networks (traffic classification)	None	ML classification
This work	IoNT (adaptive scenarios)	DNA-based adaptive crypto	XGBoost regression

Table 2 Limitations of existing works and differences with the proposed approach

Ref.	Main limitation	Differences
[17]	No network-level evaluation	No integration with IoNT dynamics
[18]	Limited scalability analysis	No ML-driven adaptation
[19]	Not nano-scale oriented	No IoNT network integration
[6]	Static configuration	No adaptive parameter tuning
[23]	Detection-only approach	No cryptographic control
[24]	No optimisation mechanism	Focus on detection, not adaptation
[26]	No security layer	No security–performance integration
This work	Simulation-based validation	Integrated adaptive security–performance framework

collecting information from the nano-network, inferring optimal encryption parameters using an ML-based robustness predictor, reporting the activity of the IoNT system, and possibly taking actions in response to the outcomes of the monitoring activities; the smart device is usually connected to the Internet.

- The *nano-routers*, which are in charge of forwarding data, when received by other nano-routers or nano-devices, to other nano-routers or to the smart device. Information exchange occurs via electromagnetic waves.
- The *nano-devices*, which are able to sense information from the environment where they are placed and, eventually, actuate some actions in response to specific commands received by nano-routers. Nano-devices send the acquired data, following a hop-by-hop schema, to nano-routers via electromagnetic waves, which guarantee very high transmission throughput (i.e., in the order of Tbps) [29]; hence, nano-devices encode messages using electromagnetic waves in the terahertz band [30].
- The *nano-controllers*, which are particular nano-devices that transmit and receive information by means of molecular communications; their main goal is to add control on the reliability of nano-devices' behavior, as will be explained in Sect. 3.2.3.

More in detail, nano-devices are equipped with nano-scale components, able to perform basic and specific tasks at the nano-level, such as sensing simple information, actuating simple actions, computing basic operations (e.g., simple cryptographic functions), storing data with limited memory capacity, and communicating over short range.

An IoNT system should be autonomous: once deployed, nano-devices must be able to self-act inside the environment. The environment hosting the nano-network should be closed

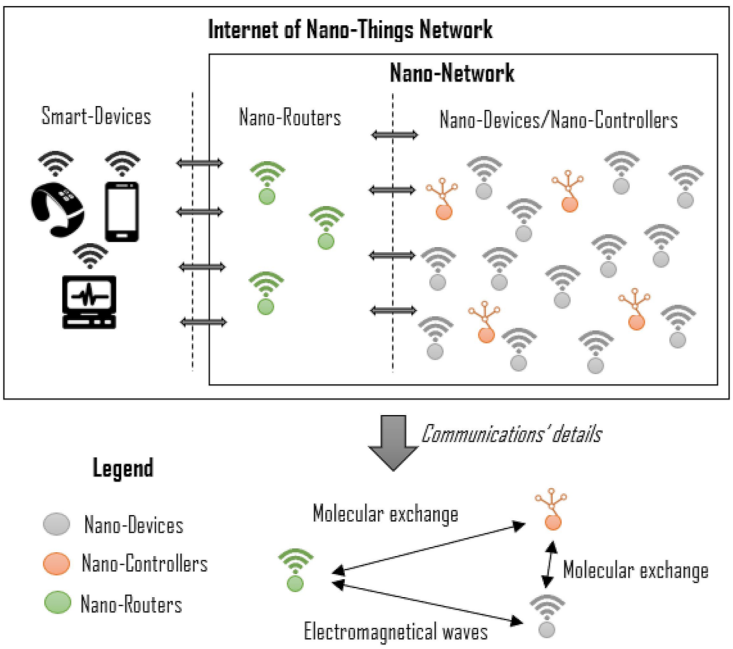


Fig. 1 Reference IoNT architecture, showing the interaction among nano-devices, nano-routers, nano-controllers, and the smart device, and highlighting the coexistence of electromagnetic and molecular communication mechanisms [3]

(i.e., the network area must be delimited). It is worth noting that, thanks to the nano-scale size of the devices, they can be massively deployed in a non-invasive way across a wide range of biological or environmental contexts, such as the human body.

The adoption of simulation tools is fundamental for early prototyping and assessing protocols and applications of such systems, since research is impaired by the unavailability of ready-to-use hardware. In the literature, several simulators are available, such as (i) *BitSimulator* [31], which is targeted to electromagnetic nano-networks; (ii) *N4Sim*[32], which is targeted to molecular communications; (iii) *Nano-Sim* [33], which is integrated within the *NS-3* simulation framework [34], and has been originally conceived for modeling baseline IoNT architectures, where the communication among nodes is handled, at the nano-scale, by means of electromagnetic waves generated in the terahertz band. In this work, we exploited the simulation environment *Nano-Sim*, whose functionalities will be detailed in the next section.

3.1 *Nano-Sim* and routing strategies

Nano-Sim identifies each nano-component by means of a unique *Dev-ID*. The most important entities are as follows:

- *Message Processing Unit*. The task of generating and processing messages is delegated to the *Message Processing Unit*. Such a module periodically (i.e., at a constant bit rate) creates packets of fixed length, customizable by the developer. Each packet also contains some headers, including (i) the *Dev-ID* of the owner/source of the data; (ii) the *Dev-ID* of the sender; (iii) the *Dev-ID* of the nano-device, which represents the next hop; (iv) an identifier of the packet (i.e., *Pack-ID*); and (v) the *Time To Live (TTL)*.
- *Physical Interface (PHY)*. Once a new message is created, the *Message Processing Unit* sends it through the protocol stack towards the *PHY*.
- *MAC (Media Access Control) Layer*. The packet is sent by the *PHY* layer to the *MAC Layer*.
- *Network Layer*. When a new packet is successfully received by the network channel through the *MAC Layer*, it is further delivered to the *Network Layer* that will check the destination of the message itself. Then, the *Network Layer* routes the packet to the final recipient.

Because nano-devices communicate with the *PHY* layer over very limited transmission ranges, multi-hop transmissions between the sender and recipient are required. As a consequence, the adopted routing protocol must efficiently manage the multi-hop connections. *Nano-Sim* provides a flexible *Network Layer*, which supports two possible routing strategies:

- The *Selective Flooding Routing (SFR)* implies the sharing of the received packet with all the devices within its transmission range. To avoid bandwidth waste, it is important to avoid duplicates. Hence, every message can be uniquely identified by the couple (*Pack-ID*, source *Dev-ID*); then, if each node keeps in memory the couples (*Pack-ID*, source *Dev-ID*) associated with the last *n* messages received, each node will not be able to transmit packets already received.
- The *Random Routing (RR)* randomly selects the next nano-device from a set of neighbors. Hence, each node knows the devices within its transmission range; for such a reason, it must cooperate with a specific MAC strategy, which is *Smart-MAC*, in this work. Also, the *RR* algorithm stores the information related to the last packets received, in order to avoid loops within the nano-network.

Both routing protocols (i.e., *SFR* and *RR*), to speed up message delivery to the smart device, require that nano-devices try to send the information to a nano-router, if available, as the next hop. Instead, due to nano-devices' energy constraints, the most promising modulation scheme for nano-communications is the *Time Spread On-Off Keying (TS-OOK)* [35]. TS-OOK offers two important advantages: (i) it does not require synchronization of nano-devices before starting packet transmission; (ii) it allows sharing among multiple devices. TS-OOK parameters, such as signal duration, transmission, frequency, and pulse energy, can be optimized based on the application scenario, as explained in Sect. 5.1. It is worth noting that transmission techniques for nano-networks are usually based on impulse communications; hence, asynchronous MAC algorithms are the best candidates for nano-networks, as they allow packet transmission without nodes competing for the channel. Furthermore, thanks to the low computing requirements, asynchronous MAC schemes are easier to implement on nano-devices with limited resources. In line with such premises, two different asynchronous strategies have been defined at the MAC level:

- The *Transparent-MAC* transmits packets from the *Network Layer* to the *Physical Interface (PHY)* without performing any kind of check.
- The *Smart-MAC* does not immediately transmit the packets from the *Network Layer* to the *PHY*, but it stores them in a dedicated queue; before starting the transmission, the MAC layer takes advantage of a handshake procedure to discover all the nano-devices available in the transmission range of the nano-device owing the message. If at least one nano-device is found, the package is then sent to the *PHY*. If the *RR* algorithm is active and the next hop has not been selected yet, the choice will be delegated to the MAC layer. In case the sending nano-device has no "neighbors," the MAC layer will apply a random back-off

delay before restarting the handshake procedure. Such a delay is uniformly distributed in the interval *(MIN back-off time, MAX back-off time)*, according to the application context.

3.2 Security functionalities

The work presented in [6] proposes the integration of security functionalities, which are based on bio-molecular cryptography. This section discusses the details concerning the DNA cryptography approach, the molecular communications, and the threat model.

3.2.1 Bio-molecular cryptography

Bio-molecular cryptography is based on the use of biological information from living beings, which is unique and therefore suitable for encrypting sensitive information. The solution proposed in [6] focuses on the use of *Deoxyribose Nucleic Acid (DNA)* and its biological properties. Performing DNA-based bio-molecular encryption requires parallel processing and bio-molecular computing capacity in order to convert short messages from a hexadecimal numeric system or from ASCII code. This work considers the bio-molecular properties of DNA sequences with the aim of generating keys that protect sensitive data contained in a packet, transmitted within the reference IoNT network. DNA processing techniques were conceived thanks to the work of Leonard Max Adleman [36]. The different phases are set up to manipulate and encode the data, starting from the alignment of DNA with other biological languages up to the protein-building process. Protein formation from sequences of DNA goes through a well-defined process, and includes the addition of several catalysts and enzymes, while still ensuring the DNA integrity. DNA represents a double helix sequence of nucleotides, which determines the code of each gene, made up of four basic elements: *Adenine (A)*, *Guanine (G)*, *Cytosine (C)*, *Thymine (T)*. Although there are only four basic elements in the sequence, their arrangement is purely random, and billions of combinations exist. The calculation, made using a DNA sequence, is called *DNA Computing*. Several issues, such as support for sending text, communicating, and video simultaneously [37], have been addressed through the use of parallel computing methods. James Dewey Watson [38] has combined traditional cryptography techniques with DNA sequences and introduced the concept of hybrid security, which can now be extended to the nano-technology domain.

The complementary strands that generate the DNA are triplets of nucleotide codons represented as follows:

```
AGG_CTC_AAG_TCC_TAG
TCC_CAG_TTC_AGG_ATC
```

If DNA strands are, as usual, mapped to numbers, alphabetic letters, or other attributes, they can be used for performing the coding and decoding of information, as well as for acting as digital data storage [39].

3.2.2 DNA manipulation

The DNA synthesis process involves a transition from DNA to *mRNA*, ultimately leading to proteins. When two DNA strands are separated by an enzyme, a new strand, named *Messenger RNA (mRNA)*, is formed, and it is complementary to the DNA strand. The RNA filament is formed by mapping DNA sequences (A, T, C, and G) to their complementary RNA sequences, which consist of U (uracil), A, G, and C [40]. Figure 2 shows how basic DNA elements are copied in an mRNA strand by replacing codon T with codon U. Finally, the combination of codons belonging to the mRNA determines the amino acids' order to build a protein cell, whose scheme is sketched in Fig. 3.

Information is secured using the DNA sequence that characterizes the biological material containing the nano-device, by mapping the message to the basic elements of DNA, properly configured. In the envisioned algorithm, a DNA strand is simulated by means of the following mapping between binary code and portions of the DNA strand:

- 00 → *"first portion of DNA strand"*
- 01 → *"second portion of DNA strand"*
- 10 → *"third portion of DNA strand"*
- 11 → *"fourth portion of DNA strand"*

The four different DNA portions are assigned by the nano-network, taking at each time different portions of the currently involved DNA sequence; in this way, when new data or a data set must be transmitted, different portions are selected. Their nature depends on the particular IoNT context considered. In fact, if the analyzed field is healthcare, we can assume that the sequences can be directly extracted from individuals and remain distinct for all the people monitored

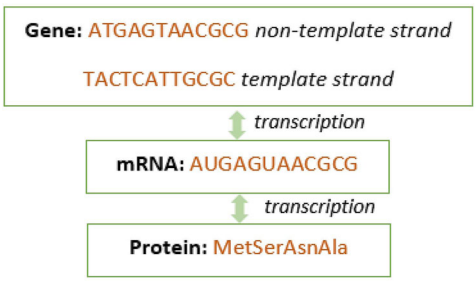


Fig. 2 Transformation process from DNA sequences to mRNA and subsequent mapping to amino acids, illustrating the intermediate steps of the bio-molecular encoding process used for data protection

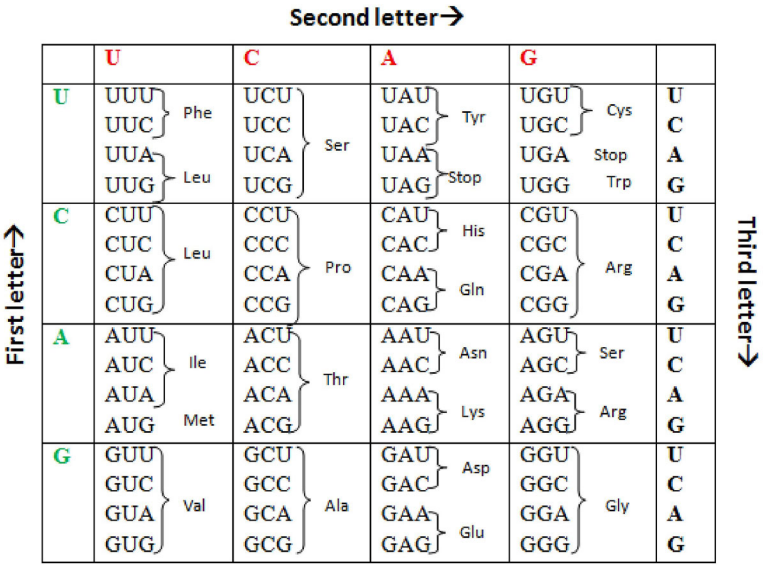


Fig. 3 Mapping between mRNA codons and amino acids, used as part of the DNA-based encoding process to transform biological sequences into structured representations for encryption

by the same healthcare structure. Note that it is important that such sequences are known to those who must decrypt the information. Other fundamental aspects in the encryption process are (i) the length of such sequences, since it affects the robustness of the security algorithm (i.e., it represents the length of the encryption key), and (ii) the two seeds (i.e., $seed_1$ and $seed_2$) owned by each nano-device.

Several steps are required before reaching the ciphertext:

1. The first step of the algorithm involves acquiring sensitive data, converting it to ASCII code, and subsequently to binary code [41].
2. Once the binary code B_{code} is generated, the DNA elements will be used to convert the message into usable DNA sequences, starting from the aforementioned seeds.
3. When a DNA portion P_{DNA} is extracted, then it is mapped with a corresponding DNA sequence P'_{DNA}, taken from a pre-defined map (containing a set of possible combinations of DNA portions with the same length of P_{DNA}), associated with the current nano-network, in the position [$seed_1$, couple of binary values (i.e., 00, 01, 10, 11) from $B1_{code}$]. The two seeds are in the range [0: ($length_{P_{DNA}}$)!] and we assume that P_{DNA} and P'_{DNA} have the same length, even if this feature is not strictly mandatory. A scheme of the map is depicted in Fig. 4.

P_{DNA}	00	01	10	11
AAAA...	TGGA...	CTAA...	...	...
GAAA...	ACGT...	...	...	...
...	...	...	...	...

Fig. 4 Example of DNA mapping scheme used for encryption, showing the association between binary values, seed parameters, and selected DNA portions for generating the encoded sequence

4. The subsequent DNA portion, namely $P2_{DNA}$, is mapped to the sequence positioned in [$seed_2$, couple of binary values $B2_{code}$], and so on, until the end of B_{code} sequence.
5. The two seeds change each time a data or a set of data which must be encrypted, by increasing their value of one unit $\%\ length_{P_{DNA}}$, in order to fit the range of the columns' number in the map.
6. Once the DNA sequence is obtained, the biological synthesis can begin, which includes (i) transformation of DNA into mRNA and (ii) mapping of the mRNA strand into amino acids according to Fig. 5.

Once the last step is completed, the set of amino acids, which contribute to the formation of proteins, is obtained. In order to complete the encryption of the obtained message, other status changes are needed: (i) conversion of the message formed by amino acids into ASCII code and (ii) conversion to binary code. The result is the encrypted message, which is now ready for its transmission towards the nano-routers, along with the information associated with the nano-controllers (see Sect. 3.2.3). No decryption is needed within the IoNT network itself; following an end-to-end approach, it guarantees the confidentiality, integrity, and availability of the transmitted information from its acquisition by a nano-device to its reception by the smart device. Figure 6 shows the flowchart for the status changes of the encryption algorithm and the reverse-decoding process.

3.2.3 Interaction with nano-controllers

Molecular exchanges do not suffer from proximity attacks, since only molecular reactions occur, not electromagnetic waves. Hence, nano-controllers, which interact with nano-routers and nano-devices through molecular exchanges, are

Ala	A	GCU, GCC, GCA, GCG	**Leu**	L	UUA, UUG, CUU, CUC, CUA, CUG
Arg	R	CGU, CGC, CGA, CGG, AGA, AGG	**Lys**	K	AAA, AAG
Asn	N	AAU, AAC	**Met**	M	AUG
Asp	D	GAU, GAC	**Phe**	F	UUU, UUC
Cys	C	UGU, UGC	**Pro**	P	CCU, CCC, CCA, CCG
Gln	Q	CAA, CAG	**Ser**	S	UCU, UCC, UCA, UCG, AGU, AGC
Glu	E	GAA, GAG	**Thr**	T	ACU, ACC, ACA, ACG
Gly	G	GGU, GGC, GGA, GGG	**Trp**	W	UGG
His	H	CAU, CAC	**Tyr**	Y	UAU, UAC
Ile	I	AUU, AUC, AUA	**Val**	V	GUU, GUC, GUA, GUG
start		AUG, GUG	***stop***		UAG, UGA, UAA

Fig. 5 Lookup table associating amino acid representations with corresponding encoded values, enabling the conversion between biological sequences and binary data within the encryption process

responsible for improving the reliability of the nano-network over time. Nano-controllers are configured for reacting in the presence of nano-devices and nano-routers [42–44]. More specifically, each nano-controller is randomly assigned within the nano-network and is configured, before network deployment, with a unique identifier id_{nc}. Nano-controllers periodically generate a concatenation between the identifier and a sequence number seq, along with a lightweight symmetric key (key_{nc}). Such a couple of information (i.e., $[id_{nc}seq, key_{nc}]$) is encoded and transmitted by the nano-controller itself and can only be decoded by nano-routers and nano-devices. In the former case, the nano-devices will use the gathered information to add a security token to the data before sending it to the nearest nano-router. In the latter case, the nano-routers will use the received token for assessing the reliability of the data received by the nano-devices. Further details on such a mechanism are available in [3]. As explained in Sect. 4, the periodicity of this molecular exchange is set by the smart device each time the ML-based robustness optimisation is taken out.

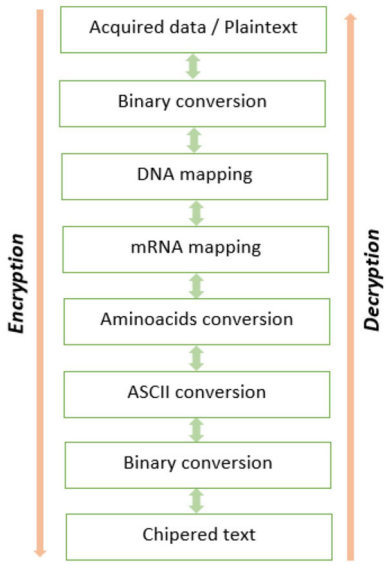


Fig. 6 Flowchart of the encoding and decoding processes, illustrating the sequence of transformations from plaintext to DNA-based encrypted data and the reverse reconstruction at the receiver side

3.2.4 Security requirements and threat model

The proposed framework is designed to achieve the following goals in terms of data protection:

- **Confidentiality of sensitive data during intra-body transmission.** Payload confidentiality relies on the DNA-based bio-molecular encryption process, which converts the original message through multiple transformations involving ASCII/binary conversion, selection of DNA portions, mapping through seed-dependent rules, transcription into mRNA, amino acid conversion, and a final binary re-encoding before transmission. As a consequence, the adversary observing only the transmitted ciphertext does not have direct access to the plaintext and, without knowledge of the selected DNA portions, the mapping table, and the seed evolution, the reverse reconstruction process becomes significantly harder than simple packet inspection. This is particularly relevant in the considered IoNT setting, where packet sniffing may reveal traffic existence but should not reveal the semantic content of the protected payload. With this respect, in order to pursue a brute-force attack, the attacker must reconstruct the selected DNA portions, the mapping rules between binary values and DNA sequences, and the evolution of the seed values used during the encoding process. Since multiple transformations are involved and the search space increases with the DNA strand length and the number of encoded blocks, exhaustive exploration becomes increasingly difficult as the configuration size grows. Also, the knowledge of some plaintext–ciphertext pairs does not directly reveal the internal DNA mappings or the seed evolution logic, while the multi-stage trans-

formation process increases the complexity of relating plaintext variations to the final ciphertext, thus increasing the resistance against chosen-plaintext attacks.

- **Integrity protection against packet manipulation and authenticity support through the molecular control token mechanism.** Integrity is supported by the end-to-end protection logic and, more specifically, coupling encrypted payload transmission and the molecular control mechanism generated by nano-controllers. In fact, nano-controllers periodically transmit encoded control information comprising an identifier, a sequence value, and a lightweight key, which can be interpreted by legitimate nano-devices and nano-routers and used as a security token for data transmissions. Such a mechanism strengthens the receiver-side ability to assess whether the received information is consistent with the expected control context, thereby making undetected spoofing or arbitrary packet injection more difficult. Although this does not constitute a full cryptographic message authentication code in the conventional sense, it provides a lightweight authenticity and reliability signal that is consistent with the constraints of the nano-scale setting.
- **Replay and MITM resistance on the communication path between nano-routers and the smart device through protected key establishment.** For the communication segment involving nano-routers and the smart device, the framework relies on a protected key negotiation process based on a certified Diffie-Hellman scheme and a subsequent key derivation function, as described in [3]. Hence, the smart device and the nano-routers can compute the session key using an ordinary/standard *Key Derivation Function (KDF)*, thereby preventing both replay and MITM attacks. This design choice improves resistance to replay and MITM attempts on that portion of the system by enabling both endpoints to derive a valid session key before exchanging information. In contrast, inside the nano-network, the very constrained nature of nano-devices makes stronger interactive authentication schemes less practical; therefore, security is pursued through payload obfuscation, limited exposure of decryption logic, control-token support, and fault-tolerant forwarding rather than through heavyweight mutual-authentication procedures at every hop.
- **Resilience against partial node failures by relying on fault-tolerant routing behavior.** The proposed framework does not fully hide communication metadata. An attacker may infer the presence of activity (i.e., packets' exchanges), estimate packet frequency, and observe message lengths or congestion conditions. Availability is challenged in IoNT environments by both malicious interference and the intrinsic fragility of nano-devices. The proposed architecture only partially addresses such an aspect. On the one hand, the routing logic provides fault tolerance because packets can still be forwarded through neighboring devices when some nano-nodes deplete resources or become unavailable. On the other hand, the framework is not designed as a full anti-jamming or anti-DoS solution, and dense deployments combined with control and handshake overhead may still experience significant performance degradation.

Because the system operates in a highly resource-constrained context, the security objective is not to eliminate all possible threats, but to raise the effort required for successful attacks while preserving an acceptable communication performance profile.

The first attack surface is the electromagnetic channel used among nano-devices and nano-routers, where short-range transmissions may be exposed to eavesdropping, spoofing, packet injection, and interference. A second attack surface is represented by the communication path between nano-routers and the smart device, which, if left unprotected, could enable session hijacking, MITM, or replay attempts. A third attack surface concerns metadata leakage: even when payloads are encrypted, an attacker may still observe packet counts, sizes, and timing. Finally, a fourth attack surface concerns the adaptive behavior of the framework itself, since an attacker could attempt to alter network conditions to indirectly influence the ML-based parameter selection process, for example, by creating congestion patterns that lead the system toward less demanding configurations.

It is fundamental to consider an adversary operating in the proximity of the IoNT deployment area and attempting to compromise the confidentiality, integrity, authenticity, or availability of the transmitted information. In line with the architecture described in Sect. 3, the attacker may target: (i) electromagnetic communications among nano-devices and nano-routers; (ii) signaling exchanges involving the smart device and nano-routers; and (iii) the correctness of the information flow used to support adaptive security decisions. The molecular channel used by nano-controllers is considered less exposed to conventional wireless interception because it is based on molecular interactions rather than radiated electromagnetic signals, although it may still be indirectly targeted through disruption or desynchronization attempts.

The attacker is assumed to be computationally bounded, but capable of wireless observation and active interference on the electromagnetic channel. More specifically, the adversary may (i) eavesdrop on packet transmissions; (ii) inject forged packets; (iii) replay previously observed packets; (iv) attempt man-in-the-middle (MITM) interaction on wireless exchanges; (v) perform traffic analysis to infer transmission rates, packet timing, and packet size; and (vi) induce congestion or service degradation through excessive packet generation or protocol manipulation. The attacker is not assumed to compromise the smart device, which is con-

sidered a trusted endpoint for decryption and control logic execution. Nano-routers are trusted to execute the communication logic and the interaction with the smart device according to the protocol. Nano-devices are assumed to be legitimate participants at deployment time, although some of them may fail, become unreachable, or be imitated by an adversary attempting spoofing actions. Nano-controllers are assumed to generate valid identifiers, sequence values, and lightweight keys before deployment, and legitimate nano-routers/nano-devices are assumed to be able to decode these control messages and exploit them to verify data reliability. The initial knowledge required for decryption, namely the DNA-related mapping material and the logic for seed evolution, is assumed to be securely provisioned to the legitimate receiver outside the nano-network.

Even if a formal cryptanalysis and the consequent attack-driven validation are out of the scope of this paper, the following main reliability-related assets are considered: (i) the confidentiality of sensed data generated by nano-devices; (ii) the integrity of packets forwarded inside the nano-network; (iii) the authenticity and freshness of control information generated by nano-controllers; (iv) the correctness of the DNA-based cryptographic mapping parameters, including DNA portions and seeds; (v) the availability and operational continuity of the IoNT communication process. Since the framework adopts an end-to-end protection approach, data are protected from the time when they are sensed by nano-devices until they are received and processed by the smart device in the external protected environment.

3.2.5 Computational complexity

The robustness of the proposed DNA-based encryption scheme relies on multiple parameters, including the selection of DNA portions, the mapping rules between binary values and DNA sequences, and the use of dynamic seed values. Assuming a DNA strand of length $\mathcal{L}_{\text{DNA}}$, the number of possible subsequences that can be selected for encoding grows combinatorially. Furthermore, for each encoding step, the mapping between binary pairs and DNA portions depends on seed values that evolve dynamically during the encryption process. If we denote by N the number of possible DNA portion selections and by S the number of possible seed values, the effective key space can be approximated as $N \times S$ for a single mapping step and increases exponentially with the number of encoded blocks.

In addition, the encryption process involves multiple transformation stages (i.e., binary encoding, DNA mapping, mRNA transcription, amino acid conversion, and re-encoding), as explained in Sect. 3.2. Without knowledge of the selected DNA portions and the seed evolution, an attacker would need to reconstruct all intermediate transformations and explore a large space of possible mappings. Therefore, the computational complexity of a brute-force attack grows exponentially with the message length and the DNA strand size, making direct inversion of the cipher computationally infeasible under realistic assumptions.

4 Machine learning functionalities

To extend the utility of the proposed IoNT framework, we introduce a ML component designed to estimate the resulting network performance and statistical robustness of the generated ciphertext, as presented in Sect. 3.2.2. In order to train the ML component, we need to generate a dataset that permits the study of the correlations between the DNA algorithm's parameters and the resulting overhead/cryptographic resilience. Note that, since publicly available datasets covering both IoNT networking dynamics and DNA-based security mechanisms are not yet available in literature, a synthetic dataset is generated through systematic simulations. More specifically, having fixed most of the simulation parameters as stated in Sect. 5, the initial DNA strand length, $\mathcal{L}_{\text{DNA}}$, is varied along with the size of the time interval between control packets emitted by the nano-controller in the molecular channel, $\mathcal{I}_{\text{mol}}$. Different variations are made, and the resultant network overhead and ciphertext randomness are measured.

More specifically, let $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N}$ be the generated dataset through a systematic parameter sweep.

- The **feature Vector (x)** or input space is defined by the primary hyperparameters governing the bio-molecular security layer. Namely,

  $$\mathbf{x} = [\mathcal{L}_{DNA}, \mathcal{I}_{mol}]$$

  where $\mathcal{L}_{DNA}$ is the initial DNA strand length and $\mathcal{I}_{mol}$ is the emission interval of molecular control packets. Note that $\mathcal{L}_{DNA}$ is varied in steps of 3 to maintain biological consistency.
- The **target Vector (y)** consists of the concatenation of the metrics that define the resulting performance and robustness:

  $$\mathbf{y} = [D_{enc}, D_{backoff}, D_{hand}, R_{av}, R_{ent}]$$

  Where D_{enc}, $D_{backoff}$, and D_{hand} indicate the average per-node delay of the encryption, backoff, and MAC-handshake procedures, respectively. Instead, R_{av}, and R_{ent} are the resulting avalanche effect [45] and Entropy of the generated ciphertexts; their computation is explained in the next Sect. 4.1.

Summarizing, the ML workflow can be divided into four phases: (i) simulation-based dataset generation, (ii) extrac-

tion of network and security features, (iii) offline model training, and (iv) online prediction-driven security adaptation executed at the smart-device level. Note that the predictive model, currently deployed at the smart device, can be the first step toward an autonomous IoNT network, where nano-controllers can dynamically configure the framework using different control actions. Secondly, the latency requirements of critical e-health applications (e.g., real-time cardiac monitoring) could benefit from this kind of predictive-based control, as the system can generate prompt responses.

4.1 Randomness measurement

Measuring the statistical randomness of the resulting cryptographic framework involves tracking every plaintext/ciphertext pairing generated during a simulation. Namely, for each pair of parameter configurations, $[\mathcal{L}_{\text{DNA}}, \mathcal{I}_{\text{mol}}]$, two simulations are made, where the unique difference is that the bitstream corresponding to the plaintext of every message differs by 1 bit.

Formally, suppose that each simulation generates M messages. Define $\mathcal{M}_A \triangleq \{m_0^A, m_1^A, ..., m_{M-1}^A\}$, and $\mathcal{M}_B \triangleq \{m_0^B, m_1^B, ..., m_{M-1}^B\}$ as the bit-encoded message set for simulations A and B, correspondingly. Note that $|\mathcal{M}_A| = |\mathcal{M}_B| = M$ and that the messages differ only by one bit. To grasp a measure of the *avalanche effect* of our framework, the mean Hamming Distance between each pair of corresponding ciphertexts is measured:

$$R_{av} = \frac{1}{M \cdot L} \sum_{i \in \{0,1,...,M\}} \text{Ham}(\mu_i^A, \mu_i^B) \tag{1}$$

where the operator $\text{Ham}(\cdot, \cdot)$ indicates the Hamming distance, μ_i^A and μ_i^B are the ciphertexts of m_i^A and m_i^B, and L is the bit-length of the messages.

Besides the relative randomness of minimal alterations of plaintext, the mean entropy of the ciphertexts is generated in our simulations. Entropy serves as a measure of the absolute randomness or unpredictability inherent in the ciphertexts, independent of pairwise comparisons. Given that the final ciphertext is binary-encoded (following the conversion from amino acid sequences to ASCII and then to binary, as detailed in Sect. 3.2.2), the most sound encoding level for computing entropy is the binary level. This aligns with the transmitted data format in the nano-network, where cryptographic resilience is ultimately measured by the difficulty of predicting or reverse-engineering the bitstream.

Formally, for a binary ciphertext μ_i of length L, the Shannon entropy per bit is computed as follows:

$$H(\mu_i) = -p_0 \log_2 p_0 - p_1 \log_2 p_1 \tag{2}$$

where p_0 and p_1 are the empirical proportions of 0 s and 1 s in μ_i, respectively. The mean entropy across all ciphertexts in a simulation (e.g., from set A) is then as follows:

$$R_{ent} = \frac{1}{M} \sum_{i=0}^{M-1} H(\mu_i^A) \tag{3}$$

A value of R_{ent} close to 1 indicates high randomness (uniform distribution of 0 s and 1 s), which is desirable for cryptographic strength.

It is worth noting that the amino acid mapping, as reported in Fig. 5, introduces potential biases, as certain letters appear more frequently in amino acid names, which may propagate through the ASCII and binary conversions, resulting in slightly lower entropy than an ideal uniform distribution. However, our algorithm prioritizes the bio-molecular foundation to enable envisioning ultra-fast implementations on biological nano-scale computation substrates. Future work could explore entropy at intermediate levels (e.g., amino acid sequences) to isolate the biological legitimacy impacts, leaving the binary level as the primary focus for end-to-end security assessment.

4.2 Learning goal

Given the tabular nature of the dataset and the expected nonlinear interactions between molecular diffusion rates ($\mathcal{I}_{mol}$) and cryptographic processing (D_{enc}), the regression task is faced using eXtreme Gradient Boosting (XGBoost) [46], due to the algorithm's efficiency in handling multi-dimensional regression tasks and its ability to capture complex dependencies without requiring the large-scale datasets typical of deep learning. The model minimizes a regularized objective function:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{4}$$

where l is the loss function (mean squared error), Ω is a penalty term to prevent overfitting, and f_k is the k-th individual decision tree.

The regularization term in Eq. 4 helps the predictor remain robust even when trained on a dataset of limited size (thousands of records). The regularization term $\Omega(f_k)$ penalizes the complexity of the k-th tree, specifically:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \|w\|^2 + \alpha \|w\|_1 \tag{5}$$

where T is the number of leaves, w is the vector of leaf scores, γ is a hyperparameter for the minimum loss reduction required to make a split, and λ and α are the L2 and L1 regularization coefficients on the leaf scores, respectively.

4.3 Online entropy optimisation

Once the XGBoost regressor $\mathcal{M}$ is trained on the dataset $\mathcal{D}$, it can be deployed at the smart device level to select the optimal DNA strand length $\mathcal{L}_{\text{DNA}}$ for a given fixed molecular control interval $\mathcal{I}_{\text{mol}}$ and application-specific latency target. The procedure, summarized in Algorithm 1, performs a fine-grained sweep over possible $\mathcal{L}_{\text{DNA}}$ values, predicts the resulting performance and robustness metrics, and returns the configuration that maximizes ciphertext randomness while respecting the latency constraint. This enables dynamic, predictive control of the bio-inspired security layer in real-time IoNT deployments.

Algorithm 1 Optimal DNA Strand Length Selection using the Trained Predictor.

Require: Trained XGBoost regression model $\mathcal{M}$, fixed molecular control interval $\mathcal{I}_{\text{mol}}^{\text{fix}}$, target maximum latency L_{tgt}, candidate DNA strand range $[\mathcal{L}_{\text{DNA}}^{\min}, \mathcal{L}_{\text{DNA}}^{\max}]$

Ensure: Optimal configuration $(\mathcal{L}_{\text{DNA}}^{*}, \hat{\mathbf{y}}^{*})$ maximising randomness subject to latency constraint

Generate candidate DNA lengths: $\mathcal{L}_{\text{cand}} \leftarrow \text{linspace}(\mathcal{L}_{\text{DNA}}^{\min}, \mathcal{L}_{\text{DNA}}^{\max}, N_{\text{cand}})$

Construct test input matrix: $\mathbf{X}_{\text{test}} \leftarrow \left[[\mathcal{L}, \mathcal{I}_{\text{mol}}^{\text{fix}}] \mid \mathcal{L} \in \mathcal{L}_{\text{cand}}\right]$

Predict performance metrics for all candidates: $\hat{\mathbf{Y}} \leftarrow \mathcal{M}.\text{predict}(\mathbf{X}_{\text{test}})$ ▷ $\hat{\mathbf{Y}} = [\hat{D}_{\text{enc}}, \hat{D}_{\text{backoff}}, \hat{D}_{\text{hand}}, \hat{R}_{\text{av}}, \hat{R}_{\text{ent}}]$

Compute total predicted delay: $\hat{D}_{\text{total}} \leftarrow \hat{D}_{\text{enc}} + \hat{D}_{\text{backoff}} + \hat{D}_{\text{hand}}$

Identify valid configurations satisfying the latency constraint: $\mathcal{V} \leftarrow \{i \mid \hat{D}_{\text{total}}[i] \leq L_{\text{tgt}}\}$

if $\mathcal{V} = \emptyset$ **then**

  return no feasible configuration

else

  Select index of maximum randomness among valid set: $i^{*} \leftarrow \arg\max_{i \in \mathcal{V}} \hat{R}_{\text{ent}}[i] + \hat{R}_{\text{av}}[i]$

  return $(\mathcal{L}_{\text{cand}}[i^{*}], \hat{\mathbf{Y}}[i^{*}])$

end if

By deploying this type of prediction machinery, the IoNT framework envisions a self-aware IoNT architecture in which the nano-controller can dynamically tune $\mathcal{I}_{mol}$ to compensate for network-congestion or enhance security during, for example, detected anomaly events.

5 Performance assessment

As introduced in Sect. 3, in line with the current scientific literature, the performance of the proposed IoNT framework and the effectiveness of the prediction module are evaluated through *Nano-Sim* simulator. Since *Nano-Sim* is mainly able to provide networking-related key performance indices, to assess the complexity of DNA computing-related operations, another tool has been adopted, similarly named *NanoSim*[1] Implemented in Python, it provides pre-loaded data sets, which can be freely downloaded from the tool's dedicated website and configured inside it. Instead, for modeling molecular exchanges, *N3Sim* [47] has been integrated. The robustness prediction module is implemented in Python's *xgboost* library[2] version 3.1.2. Experiments have been conducted on a MacBook Pro with the following features: (i) 2.5 GHz quad-core Intel Core i7 processor; (ii) 16 GB RAM memory; (iii) 500 GB flash hard disk; (iv) macOS Catalina operating system.

5.1 Simulated scenarios and configurations

The study considers a 30 mm-long artery with a diameter of 1 mm [48]. Nano-devices are uniformly distributed in this bounded environment. Nano-devices periodically send the acquired information to the nano-routers along multi-hop paths. IEEE 802.11 wireless technology is adopted to forward the received data from nano-routers to a smart device. Such a scenario is compliant with the e-health context, where nano-technologies have recently been widely adopted [49]. Examples of applications include drug delivery, gene delivery, molecular diagnostics, imaging, cardiac therapy, dental care, orthopedic applications, and so on. Regarding the protocol suite adopted for the electromagnetic-based communication inside *Nano-Sim*, three simulation setups are considered, combining different routing logic and MAC procedures, as summarized in Table 3, while Table 4 shows the parameter settings for two different scenarios. Note that data-link and routing strategies are baseline protocols already deployed into *Nano-Sim*.

As already presented, specific information on the complexity of DNA computing operations is obtained from a separate analysis conducted using *NanoSim*. The first phase of the *NanoSim* process consists of reading characterization; it provides a comprehensive alignment-based analysis and generates a set of read profiles, which serve as input to the next phase (i.e., the simulation stage), where DNA reactions take place. The authors of the simulator [50] found that the runtime of *NanoSim* scales linearly with the number of reads; in contrast, the length of the reference sequence affects the memory requirement. For example, *NanoSim* requires 4 min and 39 s and peaks at 120 MB of memory for processing 20,000 reads. Since we simulate a packet generation interval of 0.5 s and the simulation time is 5 s, we expect to produce 10 reads per nano-device. As a consequence, the time required to process the total amount of reads is approximately 0.14 s per nano-device, while the required storage should not exceed 0.06 MB per nano-device. The delay value guarantees the efficiency of the IoNT network, even in the presence

[1] NanoSim tool, https://www.bcgsc.ca/resources/software/nanosim.

[2] Python's *xgboost* library, https://pypi.org/project/xgboost/,

Table 3 Simulation setup

Simulation	Routing strategy	MAC strategy
Sim_1	SFR	Transparent-MAC
Sim_2	SFR	Smart-MAC
Sim_3	RR	Smart-MAC

of DNA computing operations, as shown in the following, while we assume that information is not persistently stored in nano-devices once transmitted, due to excessive storage occupancy that could be generated over time.

Each simulation is executed five times, changing the initial seed each time; the seed affects the random placement of nano-devices within the artery. An overview of parameter settings, derived from the study in [51], is reported in Table 5. For a more detailed component-level assessment, readers are referred to the authors' previous work [3, 6], which provides the evaluation of the proposed secure hybrid IoNT architecture, including nano-controller interactions, routing and MAC protocols, and the computational impact of DNA-based cryptographic operations.

5.2 Overhead introduced by cryptography

The message processing overhead includes the delay introduced by the bio-molecular cryptography algorithm. Note that the delay in generating the encrypted message is evaluated based on the average time (in milliseconds) required to execute the encryption algorithm on each nano-device. The resulting data confirm that the algorithm optimally works with the combination *Selective Flooding Routing - Smart-MAC*, as reported in Fig. 7a (i.e., simulation Sim_2 applied to scenario Sce_1). In scenario Sce_1, a peak is immediately observed, as the time required to encrypt the data is greater than that in all other simulations. Finally, simulation Sim_2 is not suitable for a system with many nano-devices, because the delay in generating the encrypted message scales exponentially with the number of deployed nodes. Now, we consider simulations Sim_1 and Sim_3. In a scenario with a reduced number of devices (e.g., Sce_1), using the bio-molecular encryption algorithm, combined with *SFR* or *RR*, and *Transparent-MAC* or *Smart-MAC*, respectively, takes approximately the same time. However, by increasing the number of nano-devices in the network (i.e., Sce_2), we immediately observe that the *RR* algorithm requires more processing time than the *SFR* algorithm. In fact, the handshake procedure applied by *Smart-MAC* requires much processing time with respect to the *Transparent-MAC* procedure, due to the fact that each nano-device, before sending a packet, must always check which other nano-devices are in their transmission range at that moment. The more nano-devices are present in the network, the longer the time required by the packet routing algorithm is, and this consequently causes a slowdown of the network.

Table 4 Simulation scenarios

Scenario	n. nano-devices	n. nano-routers	n. smart devices
Sce_1	50	10	1
Sce_2	200	50	1

Figure 7b analyzes the average delay generated by a single nano-device in the nano-network to execute the whole message processing, which includes the encryption algorithm and the transmission time. As the number of devices deployed in the network increases (i.e., Sim_2), the time each nano-device takes to encrypt the data to be transmitted also significantly increases, making the algorithm poorly performing in large IoNT networks, as the nano-device is simultaneously involved in forwarding operations. On the other hand, in a scenario with a reduced number of nano-devices, *SFR* combined with *Smart-MAC* is the best choice. Focusing on simulation Sim_1, we can see that as the number of sensors deployed in the network increases, the time each sensor requires to process data improves. Finally, simulation Sim_3 shows a proportional trend with respect to the number of nano-devices in the system; this is because *Smart-MAC* maps each device to check whether it is within the transmission range of the device that wants to transmit the packet.

To better understand the impact of key parameters on the proposed framework, we performed a sensitivity analysis on the DNA strand length $\mathcal{L}_{\text{DNA}}$ and the molecular control interval $\mathcal{I}_{\text{mol}}$, whose results are shown in Fig. 8. The parameter $\mathcal{L}_{\text{DNA}}$ directly affects the length of the DNA sequences used for encryption and, thus, the complexity of the bio-molecular cryptographic process. Increasing $\mathcal{L}_{\text{DNA}}$ leads to higher computational effort during encoding and decoding, which results in increased encryption delay. However, it also enhances the cryptographic robustness of the system. On the other hand, the parameter $\mathcal{I}_{\text{mol}}$ determines the frequency of the control signaling exchanged through the molecular channel. Lower values of $\mathcal{I}_{\text{mol}}$ correspond to more frequent control updates, enabling faster adaptation to changing network conditions. This improves the accuracy of reliability assessment and the responsiveness of the system, but introduces additional communication overhead and may negatively impact latency. Otherwise, larger values of $\mathcal{I}_{\text{mol}}$ reduce signaling overhead, but may slow down the system's ability to react to congestion or anomalies, thus affecting overall robustness. The combined effect of such parameters highlights a clear trade-off between security, responsiveness, and performance. In particular, configurations with larger $\mathcal{L}_{\text{DNA}}$ and smaller $\mathcal{I}_{\text{mol}}$ favor security and adaptability at the cost of

Table 5 Simulation parameters

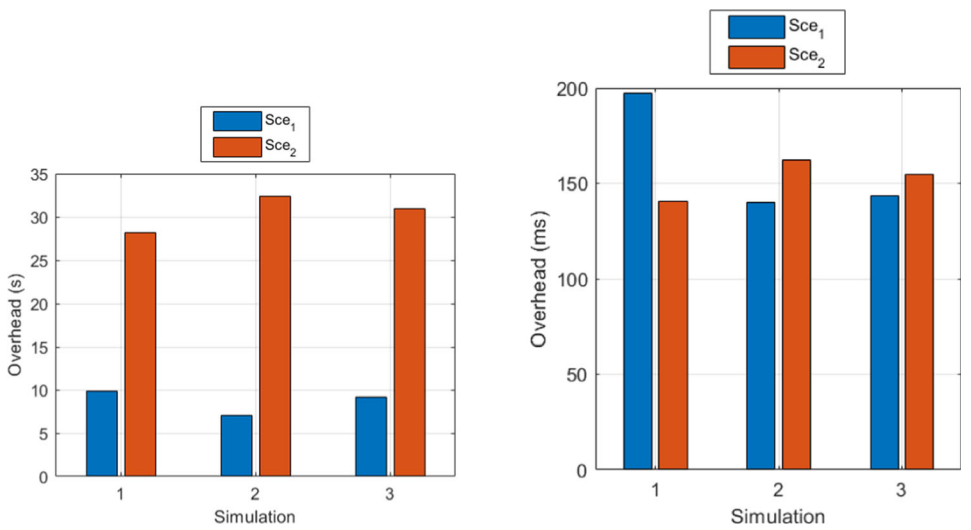
General parameters	Value
Simulation time	5 s
Number of seeds	5
Artery size	30 mm × 1 mm × 1 mm
Number of nano-devices	[50, 200]
Number of nano-routers	[10, 50]
Number of smart devices	1
Configuration of the electromagnetic-based communication channel	**Value**
Packet size	128 bytes
Packet generation time interval	0.5 s
Pulse energy	100 pJ
Pulse duration	100 fs
Pulse interval time	10 ps
Modulation scheme	TS-OOK
Backoff interval	[0 ns, 100 ns]
TTL	100
Configuration of the molecular communication channel	**Value**
Modulation scheme	OOK
Nano-controllers communication range	0.02 nm
Diffusion coefficient	1.0 nm^2/ns
Control packet generation time interval in the molecular channel	0.5 s
Brownian motion factor	0.5
Inertia factor	0.5

higher delay, whereas smaller $\mathcal{L}_{\text{DNA}}$ and larger $\mathcal{I}_{\text{mol}}$ enable lower latency, but reduced robustness. Such an analysis confirms that the proposed framework allows flexible tuning of the system behavior depending on application requirements.

Table 6 summarizes the analysis carried out in relation to the delay, considering the execution of the whole application and the average delay generated per nano-device.

5.3 Overhead introduced by ML

In order to assess the overhead introduced by the ML procedure described in Sect. 4.3, we run the simulations presented above, executing the prediction module once per second. Moreover, we trained a second dataset to represent data under network-congestion conditions. All results are averaged over



(a) Average delay of encrypted message generation

(b) Average delay of data processing per nano-device

Fig. 7 Delay analysis under different configurations: **a** impact of routing and MAC strategies on encryption delay; **b** average data processing delay per nano-device, including communication and cryptographic overhead

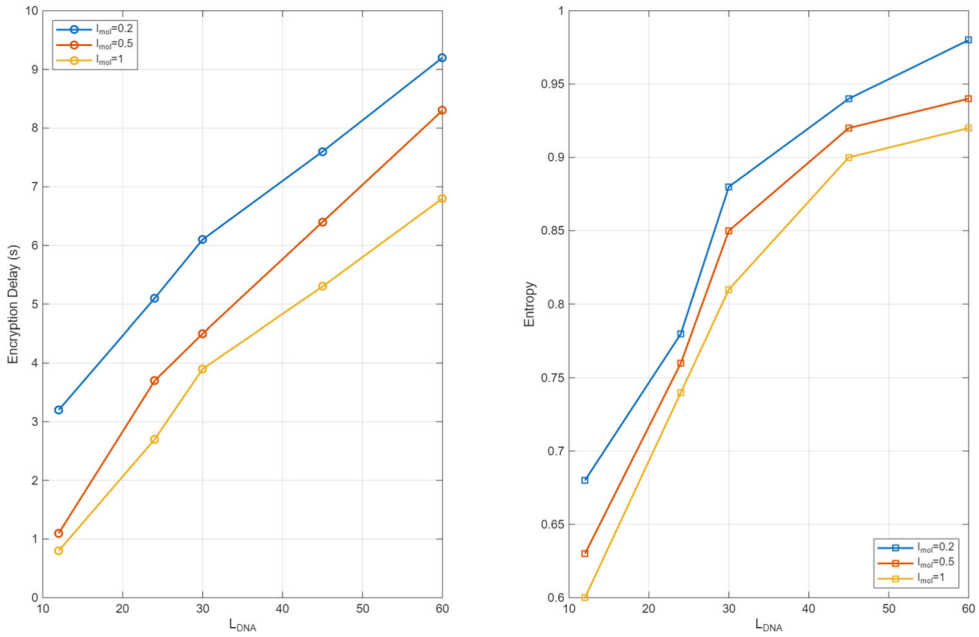


Fig. 8 Sensitivity analysis of key parameters. The left plot shows the impact of the DNA strand length $\mathcal{L}_{\text{DNA}}$ on encryption delay under different molecular control intervals $\mathcal{I}_{\text{mol}}$. The right plot reports the corresponding effect on ciphertext entropy

five simulation runs with different random seeds. To provide a more robust assessment of the observed trends, we report variability measures in terms of standard deviation, represented as error bars in Fig. 9, thus allowing to evaluate the statistical consistency of the results across different runs.

Naturally, as shown in Fig. 9, the ML procedure increases the time required to complete the simulations, but, in the presence of a congestion status, the ML algorithm allows limiting the packet losses, as reported in Fig. 10, thus improving the reliability of the network. The time spent finalizing both encryption and decryption tasks and the ML procedure inevitably increases communication latency, but they provide a significant advantage in terms of reliability for the whole system.

The observed differences among the considered configurations are consistent across multiple simulation runs, as indicated by the limited variability shown in the error bars. This suggests that the performance gap is not due to random fluctuations, but rather to the intrinsic behavior of the executed protocols.

To better isolate the contribution of each module, a comparison among the following configurations has been carried out: (i) baseline system without ML-based adaptation; (ii) system with ML-based predictive control; (iii) configurations with and without DNA-based cryptographic processing (where applicable). Such a comparison enables assessing the contribution of adaptive control and cryptographic protection in terms of delay and reliability. More in detail, this benchmarking analysis shows that the ML module mainly improves reliability under congestion, while the DNA-based cryptographic scheme primarily affects processing delay. In fact, the combined analysis reveals that the proposed framework does not uniformly outperform baseline approaches in all scenarios. Instead, it enables a tunable balance between latency and security, where the benefits of adaptive control become more evident under congestion conditions, while the cryptographic overhead becomes more significant in dense deployments.

5.4 Energy, memory, and computational cost analysis

Beyond delay-oriented metrics, it is important to discuss the implications of the proposed framework in terms of energy consumption, memory usage, and computational cost, which

Table 6 Summary of delay assessment

Sim/Delay	*Global encryption delay*	*Per nano-device encryption delay*
Sim_1	Acceptable increase, proportionally to the number of sensors	Decrease in the per nano-device incidence due to the insignificant increase in the total delay, as the devices in the network increase
Sim_2	Not suitable for a system with a high number of nano-devices	Optimal only in the case of few nano-devices
Sim_3	Acceptable increase proportionally to the number of sensors	Acceptable increase proportionally to the number of sensors

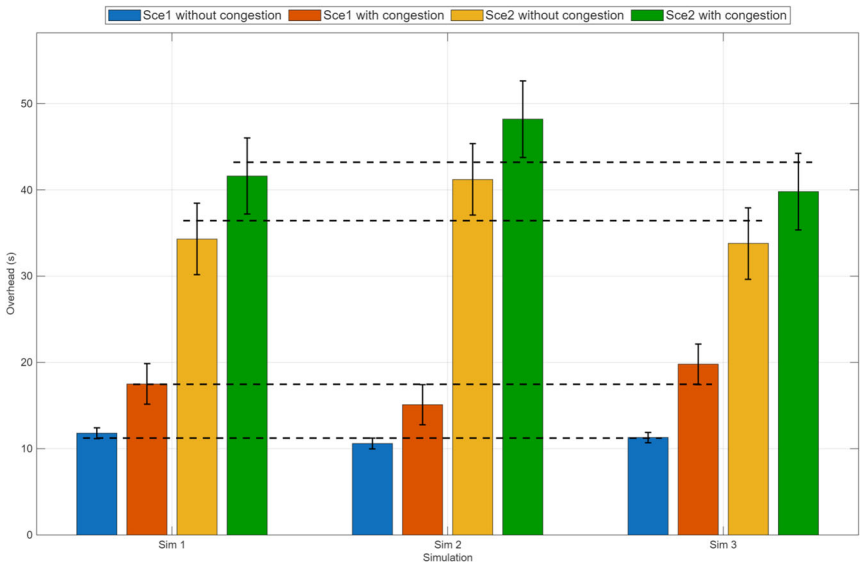


Fig. 9 Impact of the ML prediction module on overall system delay, comparing baseline configurations with and without adaptive control under congestion conditions. Error bars represent the standard deviation computed over 5 simulation runs

are critical factors in resource-constrained IoNT environments.

Energy consumption In the considered architecture, energy consumption is mainly caused by: (i) packet transmission and reception, (ii) cryptographic processing, and (iii) control signaling, including both MAC procedures and molecular exchanges. Among these, communication typically represents the dominant factor, due to the cost of electromagnetic transmissions in the terahertz band. The additional energy overhead introduced by the DNA-based cryptographic scheme is primarily due to local processing operations at the nano-device level. However, these operations involve lightweight transformations (e.g., binary-to-DNA mapping and sequence manipulation) and do not require continuous transmission; thus, their impact is expected to be lower than that of communication-related energy costs. On the other hand, Smart-MAC configurations may significantly increase energy consumption due to handshake procedures and repeated backoff operations, especially in dense scenarios. Therefore, the overall energy profile is strongly dependent on the selected routing/MAC configuration and network density.

Memory usage Memory requirements in nano-devices are extremely limited and represent a major constraint in IoNT systems. The proposed approach does not require persistent storage of large datasets at the nano-device level. Instead, each device needs to store only: (i) a limited portion of the DNA mapping table; (ii) the current seed values; and (iii) a small buffer of recently transmitted or received packets (already required by the underlying network protocols). As

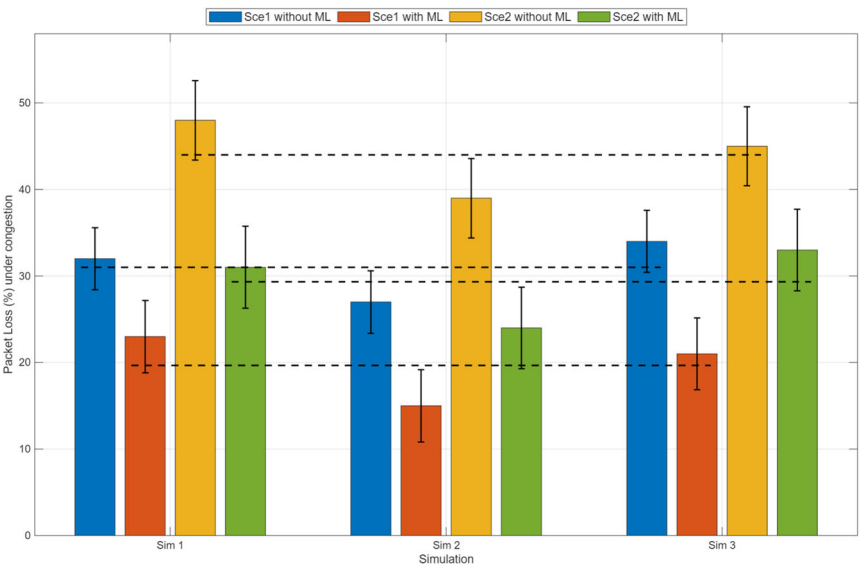


Fig. 10 Packet loss percentage under congestion conditions, showing the effect of the ML-based adaptive mechanism in improving network reliability

discussed in the simulation setup, the expected memory footprint for DNA-related processing is limited (on the order of a few tens of KB or less per nano-device), and no long-term storage of encrypted data is assumed. The ML model is deployed at the smart-device level, which is not resource-constrained, thus avoiding additional memory overhead at the nano-scale.

Computational cost The computational cost of the proposed framework can be decomposed into two main components: cryptographic processing and ML inference. The DNA-based encryption process involves a sequence of transformations whose complexity scales linearly with the length of the input message. Therefore, the computational burden remains bounded and predictable. The ML module, based on XGBoost regression, is executed by the smart device and involves inference over a relatively small feature space. Since inference in tree-based models scales with the number of trees and their depth, the resulting overhead is limited and compatible with near real-time operation. It is worth noting that no training is performed at runtime within the nano-network, thus avoiding additional computational load on nano-devices.

Such an analysis highlights that the main trade-off introduced by the proposed framework is between improved security robustness and increased processing/coordination overhead. While the cryptographic and ML components introduce additional costs, these are partially mitigated by their placement (ML at the smart device) and by the lightweight nature of the DNA-based transformations. However, system-level configurations, particularly those involving Smart-MAC and high node density, may amplify both energy and computational costs, thus limiting scalability in highly demanding scenarios.

5.5 Discussion

The obtained results highlight a clear trade-off between security-oriented control and communication efficiency in the considered IoNT framework. On the one hand, integrating DNA-based cryptography with the ML prediction module enhances the system's capability to respond to dynamic conditions and improves robustness under congestion states. On the other hand, such benefits are accompanied by additional processing and coordination costs, whose impact depends on the selected routing/MAC configuration and on the scale of the nano-network. In particular, the results show that the proposed approach reduces packet loss from approximately 44% to 29% in dense scenarios and from 31% to about 20% in smaller deployments (approximately 33–37% reduction). Such an improvement is achieved at the cost of an increased latency, with an average overhead of about 18% in dense networks and up to 55% in smaller configurations.

A first relevant observation concerns the behavior of Smart-MAC-based configurations. The increased delay observed in such cases can be motivated by the handshake procedure required before transmission, which becomes progressively more expensive as the number of nano-devices increases. Such an overhead further increases the cryptographic processing time and, in dense scenarios, may become the dominant factor affecting the end-to-end responsiveness.

A second important aspect is that the cryptographic overhead does not affect all configurations equally. In smaller-scale settings, the additional cost introduced by encryption is limited, making security-aware configurations viable when stronger protection is required. In contrast, in denser deployments, routing and medium-access interactions amplify the processing burden, thereby reducing scalability.

Finally, the results obtained under congestion conditions suggest that the ML module is beneficial when reliability is prioritized over minimal latency. In particular, the prediction-based adaptation mechanism helps reduce packet losses, indicating that dynamic parameter tuning may be preferable to static security settings in volatile IoNT environments. This also confirms that the scope of the ML module is not merely predictive accuracy, but it also contributes to maintaining a more stable security–performance balance during network operation.

5.6 Limitations

The envisioned approach presents some limitations concerning the conducted simulations, the exploited data, and cryptanalysis. In fact, the evaluation of the presented solution is simulation-based and does not include validation on real nano-devices or biological substrates. While simulation is currently the most viable approach for early-stage investigation in IoNT contexts, real-world deployment may introduce additional constraints that are not fully captured here. Fully operational nano-scale communication platforms, programmable nano-devices, and large-scale bio-nano experimental infrastructures are not yet widely available to the research community. For such a reason, the proposed framework has been evaluated using well-known (i.e., already validated by the research community) simulation tools specifically designed for nano-networks, namely $Nano-Sim$, $N3Sim$, and $NanoSim$.

The absence of direct benchmarking against alternative adaptive security frameworks or lightweight cryptographic schemes is due to the fact that most existing proposals target different application domains, network architectures, threat models, or evaluation environments. But, above all, no solution available in literature combines both DNA-based cryptographic approaches and ML-based mechanisms for nano-scale networks, as emerged from Sect. 2. Hence, the

present study focuses on internal benchmarking aimed at quantifying the impact of the proposed adaptive mechanisms under different routing, MAC, congestion, and cryptographic configurations.

Although the evaluation focuses on a healthcare scenario, the proposed framework is not intended to be limited to this application domain. In fact, the key components of the system (i.e., the DNA-based cryptographic module and the ML-driven adaptive control) operate independently of the specific physical environment. From a networking perspective, the considered scenario already captures several features of IoNT systems, including constrained devices, limited communication range, multi-hop routing, and dynamic network conditions. Such features are also present in other domains, such as industrial IoNT systems (e.g., smart manufacturing) and environmental monitoring deployments (e.g., distributed sensing networks). Differences across domains mainly affect parameters such as network size, node density, and traffic patterns. However, the impact of the key parameters (i.e., $\mathcal{L}_{\text{DNA}}$ and $\mathcal{I}_{\text{mol}}$) analyzed in this work remains consistent across contexts. Therefore, the proposed framework can be adapted to different IoNT deployments by tuning such parameters according to application-specific requirements, thus paving the way for further research areas.

Moreover, the ML module is trained on data generated within the same simulation framework. Although this choice is justified by the lack of publicly available real IoNT datasets, it may limit the generalizability of the predictor in environments characterized by different dynamics, noise sources, or hardware constraints. Note that the ML component is implemented using XGBoost due to its capability to capture non-linear relationships in tabular datasets, while maintaining acceptable computational complexity. The present work does not provide a systematic comparison with alternative regression techniques, such as *Random Forest, LightGBM, Support Vector Regression,* or *Neural Networks*. Therefore, the reported results do not aim to demonstrate that XGBoost represents the optimal solution: a comprehensive benchmarking campaign involving multiple learning models could be considered a future research direction in order to further optimize the outcome obtained in the investigated context.

Also, the work mainly focuses on delay-related metrics and ciphertext randomness indicators, while other important dimensions such as energy consumption, memory occupation, and low-level hardware feasibility are not yet comprehensively assessed. This is due to the capabilities of the currently available nano-network simulation tools, which do not provide sufficiently detailed models for accurately measuring energy consumption, memory occupation, or hardware-level computational costs associated with DNA-based processing. Anyway, the main goal of this work is to investigate the impact of the proposed adaptive security framework on communication reliability and delay, as such metrics are directly affected by both the DNA-based cryptographic mechanism and the ML-driven adaptation process. For such a reason, latency and packet loss have been selected as the main quantitative evaluation metrics.

Finally, the current security analysis remains primarily qualitative and does not provide formal cryptographic proofs under standard adversarial models. In particular, the framework has not been formally evaluated against chosen-plaintext attacks, known-plaintext attacks, ciphertext-only attacks, or other advanced cryptanalytic techniques. Similarly, although replay and MITM resistance are supported by the protected key-establishment procedures inherited from the reference architecture, no formal security proof is currently available. Therefore, the proposed framework should be regarded as an adaptive security architecture whose feasibility and performance have been demonstrated through simulation, rather than as a cryptographic scheme with formally established security guarantees.

6 Conclusions

The paper presented an IoNT framework that combines DNA-based bio-molecular cryptography with an ML-driven prediction module to support the adaptive configuration of security parameters under dynamic network conditions. The conducted evaluation showed that the proposed approach can improve robustness in congestion-prone settings, while maintaining acceptable performance in selected configurations, although additional latency and coordination overhead are introduced. In particular, the results indicate that the impact of the security mechanisms strongly depends on the routing/MAC strategy and on the density of the deployed nano-devices. The proposed approach also highlights that adaptive parameter tuning can be more effective than static security configurations when network conditions vary over time.

Future work will focus on four directions: (i) extending the evaluation to additional IoNT scenarios (e.g., larger-scale deployments, different traffic profiles) beyond healthcare, with the use of synthetic and real data for ML training; (ii) incorporating further metrics such as energy consumption, memory usage, and computational cost; (iii) strengthening the security analysis with more formal threat-driven validation; (iv) investigating real-world prototyping opportunities through interdisciplinary collaborations.

Author contribution Conceptualization: A.R., S.S., L.A.G., G.P., J.F.C.M., A.C.P. Methodology: A.R., S.S., L.A.G., G.P., J.F.C.M., A.C.P. Writing - Original Draft: A.R., S.S., L.A.G., G.P., J.F.C.M., A.C.P. Writing - Review & Editing: A.R., S.S., L.A.G., G.P., A.C.P. Supervision: S.S., L.A.G., A.C.P.

Funding Open access funding provided by Università degli Studi dell'Insubria within the CRUI-CARE Agreement. No funding has been received for the presented work.

Data availability No datasets were generated or analysed during the current study.

References

1. Akyildiz IF, Brunetti F, Blázquez C (2008) Nanonetworks: a new communication paradigm. Comput Netw 52(12):2260–2279. https://doi.org/10.1016/j.comnet.2008.04.001
2. El-Din HE, Manjaiah D (2017) Internet of Nano Things and Industrial Internet Of Things. In: Internet of Things: novel advances and envisioned applications, pp 109–123. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-319-53472-5_5
3. Sicari, S., Rizzardi, A., Piro, G., Coen-Porisini, A., Grieco, L (20109) Beyond the smart things: towards the definition and the performance assessment of a secure architecture for the Internet of Nano-Things. Computer Networks 162:106856. https://doi.org/10.1016/j.comnet.2019.07.012
4. Maynard AD (2006) Nanotechnology: assessing the risks. Nano Today 1(2):22–33. https://doi.org/10.1016/S1748-0132(06)70045-7
5. Niu Y, Zhao K, Zhang X, Cui G (2020) Review on DNA cryptography. In: Bio-inspired Computing: Theories and Applications: 14th International Conference, BIC-TA 2019, Zhengzhou, China, November 22–25, 2019, Revised Selected Papers, Part II 14, pp 134–148. https://doi.org/10.1007/978-981-15-3415-7_11. Springer
6. Rizzardi A, Piro G, Sicari S, Grieco LA, Coen-Porisini A (2025) Bio-molecular cryptography for protecting nano-network transmissions in healthcare applications. In: 2025 13th Wireless Days Conference (WD), pp 1–9. https://doi.org/10.1109/wd67713.2025.11302720. IEEE
7. Xiao G, Lu M, Qin L, Lai X (2006) New field of cryptography: DNA cryptography. Chin Sci Bull 51(12):1413–1420. https://doi.org/10.1007/s11434-006-2012-5
8. Anam B, Sakib K, Hossain M, Dahal K (2010) Review on the advancements of DNA cryptography. In: Proceedings of International Conference on Software, Knowledge, Information Management and Applications (SKIMA)
9. Mahjabin T, Olteanu A, Xiao Y, Han W, Li T, Sun W (2023) A survey on DNA-based cryptography and steganography. IEEE Access 11:116423–116451. https://doi.org/10.1109/ACCESS.2023.3324875
10. Chu L, Su Y, Yao X, Xu P, Liu W (2025) A review of DNA cryptography. Intelligent Computing 4:0106. https://doi.org/10.34133/icomputing.0106
11. Dressler F, Kargl F (2012) Towards security in nano-communication: challenges and opportunities. Nano Commun Networks 3(3):151–160. https://doi.org/10.1016/j.nancom.2012.08.001
12. Wendt JB, Potkonjak M (2011) Nanotechnology-based trusted remote sensing. In: Sensors, pp 1213–216. https://doi.org/10.1109/icsens.2011.6127174. IEEE
13. Zhang Y, Wang F, Chao J, Xie M, Liu H, Pan M, Kopperger E, Liu X, Li Q, Shi J et al (2019) DNA origami cryptography for secure communication. Nat Commun 10(1):1–8. https://doi.org/10.1038/s41467-019-13517-3
14. Sreeja, CS, Misbahuddin M, Mohammed Hashim NP (2014) DNA for information security: a survey on DNA computing and a pseudo DNA method based on central dogma of molecular biology. In: International conference on computing and communication technologies, pp 1–6. https://doi.org/10.1109/iccct2.2014.7066757
15. Vidhya E, Rathipriya R (2020) Key generation for DNA cryptography using genetic operators and Diffie-Hellman key exchange algorithm. Computer Science 15(4):1109–1115
16. Das A, Sarma SK, Deka S (2021) Data security with DNA cryptography. In: Transactions on engineering technologies, pp 159–173. Springer, Singapore. https://doi.org/10.1007/978-981-15-8273-8_13
17. AL-Shargabi B, Dar Assi A (2023) A modified lightweight DNA-based cryptography method for internet of things devices. Expert Syst 40(6):13270. https://doi.org/10.1111/exsy.13270
18. Aqeel S, Khan AS, Abbasi IA, Algarni F, Grzonka D (2025) Enhancing IoT security with a DNA-based lightweight cryptography system. Sci Rep 15(1):13367. https://doi.org/10.1038/s41598-025-96292-0
19. Qiqieh I, Alzubi J, Alzubi O (2025) DNA cryptography based security framework for health-cloud data. Computing 107(1):35. https://doi.org/10.1007/s00607-024-01393-9
20. Bulasara PK, Sahoo SR (2024) A robust and secure drug delivery with single transmitter and dual symmetrical receivers in an Internet of Bio-Nano Things. IEEE Internet Things J 11(14):25074–25087. https://doi.org/10.1109/jiot.2024.3393637
21. Rana A, Gautam D, Kumar A, Kumar K, Vasilakos AV, Das AK, K VB (2025) A comprehensive review of machine learning applications for Internet of Nano Things: challenges and future directions. Artif Intell Rev. https://doi.org/10.1007/s10462-025-11211-z
22. Rana A, Gautam D, Kumar P, Das AK (2024) Architectures, benefits, security and privacy issues of Internet of Nano Things: a comprehensive survey, opportunities and research challenges. IEEE Commun Surv Tutor. https://doi.org/10.1109/COMST.2024.3423477
23. Bakhshi T, Shahid S (2019) Securing Internet of bio-Nano Things: ML-enabled parameter profiling of bio-cyber interfaces. International Multi-Topic Conference. https://doi.org/10.1109/inmic48123.2019.9022753
24. Bakhshi T, Zafar S (2023) Hybrid deep learning techniques for securing bioluminescent interfaces in Internet of Bio Nano Things. Italian National Conference on Sensors. https://doi.org/10.3390/s23218972
25. Zafar S, Nazir M, Sabah A, Jurcut AD (2021) Securing bio-cyber interface for the Internet of Bio-Nano Things using particle swarm optimization and artificial neural networks based parameter profiling. Comput Biol Med. https://doi.org/10.1016/j.compbiomed.2021.104707
26. Galal A, Hesselbach X (2022) Machine learning models for traffic classification in electromagnetic nano-networks. IEEE Access. https://doi.org/10.1109/ACCESS.2022.3165013
27. Rajawatl AS, Kumar J, Goyal SB, Potgantwar A, Bedi P (2023) Deep learning and blockchain for securing 5G enabled IoNT in big data environment. Int Congress Inf Commun Technol. https://doi.org/10.1109/icict57646.2023.10134330

28. Bensalem M, Jukan A (2021) Benchmarking machine learning techniques for THz channel estimation problems. arXiv (Cornell University)
29. Akyildiz I, Jornet JM (2010) Nano communication networks. Nano Commun. Networks 1:3–19. https://doi.org/10.1016/j.nancom.2010.04.001
30. Jornet JM, Akyildiz IF (2011) Channel modeling and capacity analysis for electromagnetic wireless nanonetworks in the terahertz band. IEEE Trans Wireless Commun 10(10):3211–3221. https://doi.org/10.1109/TWC.2011.081011.100545
31. Dhoutaut D, Arrabal T, Dedu E (2018) Bit simulator, an electromagnetic nanonetworks simulator. In: Proceedings of the 5th ACM international conference on nanoscale computing and communication, pp 1–6. https://doi.org/10.1145/3233188.3233205
32. Turgut NA, Bilgin BA, Akan OB (2021) N4sim: The first nervous nanonetwork simulator with synaptic molecular communications. IEEE Trans Nanobiosci 21(4):468–481. https://doi.org/10.1109/TNB.2021.3118851
33. Piro G, Grieco LA, Boggia G, Camarda P (2013) Nano-sim: simulating electromagnetic-based nanonetworks in the network simulator 3. In: Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques, pp 203–210. https://doi.org/10.4108/icst.simutools.2013.251699. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering)
34. Piro G, Grieco LA, Boggia G, Camarda P (2013) Simulating wireless nano sensor networks in the ns-3 platform. In: IEEE 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp 67–74. https://doi.org/10.1109/WAINA.2013.20. IEEE
35. Jornet JM, Akyildiz IF (2011) Information capacity of pulse-based wireless nanosensor networks. In: 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pp 80–88. https://doi.org/10.1109/SAHCN.2011.5984951. IEEE
36. Adleman LM (1994) Molecular computation of solutions to combinatorial problems. Science 1021–1024. https://doi.org/10.1126/science.7973651
37. Azfar A, Choo K-KR, Liu L (2014) A study of ten popular android mobile voip applications: Are the communications encrypted? In: 47th Hawaii international conference on system sciences, pp 4858–4867. https://doi.org/10.1109/HICSS.2014.596. IEEE
38. Watson JD, Crick F et al (1953). A structure for deoxyribose nucleic acid. https://doi.org/10.1038/171737a0
39. Adleman LM, Rothemund PW, Roweis S, Winfree E (1999) On applying molecular computation to the data encryption standard. J Comput Biol 6(1):53–63. https://doi.org/10.1089/cmb.1999.6.53
40. Lloyd S, Snell QO (2008) Sequence alignment with traceback on reconfigurable hardware. In: International Conference on Reconfigurable Computing and FPGAs, pp 259–264. https://doi.org/10.1109/ReConFig.2008.30. IEEE
41. Kreibich C, Crowcroft J (2006) Efficient sequence alignment of network traffic. In: Proceedings of the 6th ACM SIGCOMM conference on internet measurement, pp 307–312. https://doi.org/10.1145/1177080.1177121
42. Pierobon M, Akyildiz IF (2010) A physical end-to-end model for molecular communication in nanonetworks. IEEE J Sel Areas Commun 28(4). https://doi.org/10.1109/JSAC.2010.100509
43. Atakan B, Akan OB, Balasubramaniam S (2012) Body area nanonetworks with molecular communications in nanomedicine. IEEE Commun Mag 50(1). https://doi.org/10.1109/MCOM.2012.6122529
44. Chouhan L, Alouini M-S (2023) Interfacing of molecular communication system with various communication systems over internet of every nano things. IEEE Internet Things J 10(16):14552–14568. https://doi.org/10.1109/JIOT.2023.3273030
45. Webster AF, Tavares SE (1986) On the Design of S-Boxes. In: Williams HC (ed) Advances in Cryptology — CRYPTO '85 Proceedings, pp 523–534. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-39799-X_41
46. Chen T, Guestrin C (2016) Xgboost: A scalable tree boosting system. In: KDD, pp 785–794. ACM, San Francisco, CA, USA. https://doi.org/10.1145/2939672.2939785
47. Llatser I, Demiray D, Cabellos-Aparicio A, Altilar DT, Alarcon E (2014) N3sim: Simulation framework for diffusion-based molecular communication nanonetworks. Simul Model Pract Theory 42:210–222. https://doi.org/10.1016/j.simpat.2013.11.004
48. Piro G, Yang K, Boggia G, Chopra N, Grieco LA, Alomainy A (2015) Terahertz communications in human tissues at the nanoscale for healthcare applications. IEEE Trans Nanotechnol 14(3):404–406. https://doi.org/10.1109/TNANO.2015.2415557
49. Sahoo S, Parveen S, Panda J (2007) The present and future of nanotechnology in human health care. Nanomed Nanotechnol Biol Med 3(1):20–31. https://doi.org/10.1016/j.nano.2006.11.008
50. Yang C, Chu J, Warren RL, Birol I (2017) Nanosim: nanopore sequence read simulator based on statistical characterization. GigaScience 6(4):010. https://doi.org/10.1093/gigascience/gix010
51. Funck C, Laun FB, Wetscherek A (2018) Characterization of the diffusion coefficient of blood. Magn Reson Med 79(5):2752–2758. https://doi.org/10.1002/mrm.26919