IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 13, 2022, accepted March 31, 2022, date of publication April 5, 2022, date of current version April 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3165013

Machine Learning Models for Traffic Classification in Electromagnetic Nano-Networks

AKRAM GALAL AND XAVIER HESSELBACH, (Senior Member, IEEE)
Department of Network Engineering, Universitat Politècnica de Catalunya (UPC)—Barcelonatech, 08034 Barcelona, Spain

Corresponding author: Akram Galal (akram.galal@upc.edu)

This work was supported in part by the "Agencia Estatal de Investigación" of "Ministerio de Ciencia e Innovación" of Spain under Project PID2019-108713RB-C51/MCIN/AEI/10.13039/501100011033, and in part by the "Agència de Gestió d'Ajuts Universitaris i de Recerca" (AGAUR) of the "Generalitat de Catalunya" under Grant 2021FI_B2 00091.

ABSTRACT The number of nano-sensors connected to wireless electromagnetic nano-network generates different traffic volumes that have increased dramatically, enabling various applications of the Internet of nano-things. Nano-network traffic classification is more challenging nowadays to analyze different types of flows and study the overall performance of a nano-network that connects to the Internet through micro/nano-gateways. There are traditional techniques to classify traffic, such as port-based technique and load-based technique, however the most promising technique used recently is machine learning. As machine learning models have a great impact on traffic classification and network performance evaluation in general, it is difficult to declare which is the best or the most suitable model to address the analysis of large volumes of traffic collected in operational nano-networks. In this paper, we study the classification problem of nano-network traffic captured by micro/nano-gateway, and then five supervised machine learning algorithms are used to analyze and classify the nano-network traffic from traditional traffic. Experimental analysis of the proposed models is evaluated and compared to show the most adequate classifier for nano-network traffic that gives very good accuracy and performance score to other classifiers.

INDEX TERMS Machine learning, micro/nano-gateway, nano-network, traffic classification, traffic monitoring.

I. INTRODUCTION

Nano-technology has emerged to provide new opportunities for sensing and actuating [1]. Nano-sensors can sense, compute and communicate into networks or the Internet, enabling advanced applications in different fields such as the biomedical, environmental, industrial and military fields [2]. In the biomedical field, nano-sensors are used in drug delivery and medical treatment applications [3], [4]. Likewise, nano-sensors are used in health tracking systems and human body communication, promoting medical body area networks that allow medical personnel to have remote access and monitoring to the patient's body [5]. Through wearable health trackers, doctors can monitor real-time measurements of the human body like heartbeats rate, blood pressure and breath tests to get early detection of sickness [4], [6]. In the environmental field, nano-nodes play a significant role in monitoring the spread of viruses and diseases in public locations. Also, air pollution can be controlled by utilizing nano-filters to improve the air quality and remove harmful substances. Nano-networks can support the industrial field by improving new materials, producing procedures and quality control strategies. Moreover, they are involved in the agriculture industry to deliver pesticides, enhance food quality and water control. While in the military field, nano-sensors can be used in nuclear, biological and chemical defense [6], [7].

A Nano-network is a type of network that exchanges information wirelessly between a group of nano-devices fabricated at the nano-scale [8]. These nano-devices can communicate internally with each other in the nano-domain, or externally on the Internet, forming the Internet of Nano-Things (IoNT) with the aid of smart hybrid interfaces called micro/nano-gateways. These gateways can communicate in dual communication paradigms, i.e., the nano-scale paradigm and micro/macro-scale paradigm that represents traditional communication networks [9]. Nano-devices have many constraints, such as inadequate energy resources,

The associate editor coordinating the review of this manuscript and approving it for publication was Feng Lin.

topology-unawareness, limited computational power, high sensitivity for emergency information, redundant data, limited storage and computational capabilities, besides data routing without storing routing tables in minimal transmission ranges [10]. In addition to the aforementioned constraints, interoperability and heterogeneity are the main features of IoNT, because exchanging different data formats between several communication protocols in diverse nano-network applications leads to a lack of cooperation and capability mismatch between devices, which affects the overall performance of the nano-network [11]. Hence, the micro/nano-gateway interacts between the nano-network domain and the traditional network domain by providing a protocol mapping mechanism that allows an intermediary data format conversion for various protocols of the nano-network domain to the traditional domain and vice versa. Therefore, providing access to the nano-network from the Internet and facilitating the development of different applications [11]. The two-sided communication capability of these micro/nano-gateways causes some compatibility issues, because of different communication protocols and data formats in different domains, on top of the heterogeneity of nano-devices/sensors, i.e., each device can operate with a certain communication protocol using a specific data format within a particular application, which adds more challenges to the micro/nano-gateway functionality [12]. To overcome these limitations and constraints, the micro/nano-gateway needs to allocate the main information from each packet received from nano-domain protocols and convert it to a suitable data format to enable linking the nano-network with the Internet by affording a protocol conversion mechanism to accommodate different data formats coming from various nano-sensors [11].

Moreover, a data characterization mechanism is needed to differentiate between different types of traffic, such as the high-priority and low-priority traffic coming to/from the nano-network, which results in the loss of high-priority or critical information during a high data traffic load. During high data traffic load, packets are randomly dropped and delayed, which increases the delay and packet loss of high priority data. Thus, the absence of a data categorization mechanism can affect the primary requirement of delivering high-priority information with minimal delay and loss [13]. Therefore, a data characterization mechanism and traffic classification are essential for the micro/nano-gateway to ensure traffic monitoring, Quality of Service (QoS) management and maintaining security access. Besides, traffic classification can implement a mechanism for different services that classifies the traffic flow according to the application type or according to the communication direction, whether it is from the Internet to the nano-network or vice-versa. Accordingly, computing resources can be allocated and QoS can be guaranteed, thus improving network management and overall nano-network performance. As a result, and to address these challenges, the development of novel Machine Learning (ML) methods has become necessary for nano-network traffic, where ML can assist in devising novel methods and accurate ways for better performance evaluation and traffic classification for nano-networks [14].

Recently, innovative development of new tools from the field of ML has been considered in several engineering fields, which enables the analysis of large datasets through training models. These models can be utilized for observations, classification or predictions. Computer vision, natural language processing, speech and image recognition are among these fields, in addition to telecommunication networks, such as wireless sensor networks, Internet of Things, cognitive radio networks, satellite communication, cloud/edge computing, software-defined networking and machine-to-machine networks. This frontier is continuing its expansion by including the nano-communication field, and it is expected to have a significant impact on the design of novel nano-materials, nano-scale communication networks and data-driven biomedicine applications [14].

ML has shown tremendous benefits in solving complex network problems and providing situation and parameter predictions. However, heavy resources are required to process and analyze the data that can be done either offline or using edge computing, which also requires heavy transmission rates to provide a timely response. When it comes to the nano-network, in addition to its role of providing Internet access to different nano-devices, the micro/nano-gateway represents the edge of the nano-network [11]. Similar to the progression from cloud computing to cloud intelligence, a fast evolution on the edge of the nano-network from edge computing to edge intelligence or Edge AI that will provide adaptation for data-driven applications, enhance nano-network access performance and enable the deployment of quality of experience, security and privacy targets. Despite all the promises ahead, the road to realizing Edge AI in nano-network is still in its early stages. Accordingly, the main objective of this paper is to extend the intelligence of the micro/nano-gateway to process and analyze the nano-network traffic at run time and provide timely and efficient communication in both the upstream and downstream directions [11]. This will be achieved by applying traffic classification techniques based on machine learning, as ML models embed intelligence into network functions and attract research interest because of their expected accuracy and efficiency. However, selecting the best machine learning model that fits a specific problem is not an easy task. Even if multiple models can be well fitted for a particular use case, it may be a little bit difficult to figure out the model/algorithm that provides optimal performance [15]. As a result, the focus so far is to bring intelligence to the micro/nano-gateway by studying the nano-network traffic classification problem and its interlock with ML aiming at providing Edge AI in the nano-network domain, in addition to figuring out the ML model that best fits the nano-network traffic, which represents the main motivations behind this paper. The major contributions of this paper are as follows:

- Develop a nano-network traffic generator that generates packets in nano-domain format. This traffic combined

with background traffic represents the traditional network domain to form a synthetic dataset received by the micro/nano-gateway device. The traffic received by the micro/nano-gateway is classified into four classes; i) TCP packet, ii) UDP packet, iii) nano-to-nano-communication packet (NN0) or iv) nano-to-Internet-communication packet (NN1).
- Investigate five standard supervised machine learning algorithms on the generated dataset to figure out which model is the best fit in the analysis and classification of the traffic in order to provide efficient performance for the micro/nano-gateway. These algorithms are Decision Trees Classifier (DTC), Support Vector Machines (SVM), the K-Nearest Neighbors (KNN), Random Forest (RF) and Naïve Bayes (NB).
- Evaluate the five models' performance using different metrics before and after manipulating the corresponding hyper-parameters for each model to achieve optimized tuning values, then a comparative analysis has been introduced between all models to select the best model that provides the best fit for the traffic classification issue.

The rest of the paper is organized as follows. The literature review and related work are presented in Section II. The implementation of the traffic generator and the corresponding dataset is illustrated in Section III. The experimental results of the study and the evaluation of the proposed models are illustrated in Section IV, followed by the conclusion and future work in Section V.

II. LITERATURE REVIEW AND RELATED WORK

In this section, we briefly discuss the relevant work in the literature for some approaches related to the scope of the paper, such as machine learning types, learning algorithms, performance metrics, traffic classification and prediction issues and machine learning approaches in IoT and IoNT communication domains.

A. MACHINE LEARNING TYPES

Machine learning is a subfield of computer science that is concerned with solving practical problems by gathering a dataset and building statistical models based on that dataset to solve these problems. ML enables machines to learn without explicit programming by applying some learning algorithms that take a set of samples as an input named a training set. In general, learning can be supervised, semi-supervised, unsupervised or reinforcement [16], [17].

In supervised learning, the dataset is the collection of labeled examples, and the goal of a supervised learning algorithm is to use the dataset to produce a model that takes a feature vector as input and outputs information that allows deducing the label for this feature vector. The objective of supervised learning is to learn how to predict the appropriate output vector for a given input vector. In unsupervised learning, the dataset is a collection of unlabeled examples, there are no labels required for the training set. The goal of an unsupervised learning algorithm is to create a model that takes a feature vector as an input and reconstructs it into another vector or a value that can be used to solve a practical problem. In unsupervised learning, no labels are required for the training set. While, in semi-supervised learning, the dataset contains both labeled and unlabeled examples. Usually, the quantity of unlabeled examples is much higher than the number of labeled examples. The goal of a semi-supervised learning algorithm is the same as a supervised learning algorithm, but the concern here is that using many unlabeled examples can help the learning algorithm to produce a better model. Reinforcement learning is a subfield of machine learning where the machine lives in an environment to perceive its state as a vector of features. The machine can execute actions in every state. Different actions bring different rewards and will move the machine to another state in the environment. It deals with the problem of learning the appropriate action or sequence of actions to be taken in a given situation to maximize payoff. The goal of a reinforcement learning algorithm is to learn a policy that takes the feature of a certain state as input and outputs an optimal action to execute that state. The action is optimal if it maximizes the expected average reward [16], [17].

In machine learning, when the target labels consist of a finite number of discrete categories, they are known as classification tasks, while the target labels are composed of one or more continuous variables are known as regression tasks [17].

Deep Learning (DL) is considered a specialized subset of machine learning. It is a series of algorithms founded on the Artificial Neural Network (ANN) having multiple layers, where the design of such an ANN is inspired by the biological neural network of the human brain. DL models analyze data with a logical structure similar to how humans conclude [18]. Nowadays, DL is widely used in multiple topics, having a wide range of applications such as computer vision, natural language processing, speech recognition, visual object detection, bioinformatics and biomedicine, however its applications are quite limited in the data network domain due to some reasons, such as data availability, visibility and computing power [15].

It is worth noting that DL models are data-eager. They require incredibly vast amounts of data to be trained. For instance, Tesla's autonomous driving software needs millions of images and video hours to function properly, which is difficult to apply in a nano-network that has limited processing capacity. Moreover, DL models have some problems with visibility and interpretation. Although they perform feature selection from input data and predict the output with high accuracy, it is very difficult to understand their internal functionalities, as they are like black boxes [15]. Furthermore, DL needs substantial computational power to be trained. However, with the emergence of cloud computing infrastructure and high-performance Graphic Processing Unit (GPU), the time of training a deep learning network could be reduced from weeks to hours. Although the GPU can

provide a faster calculation in the training phase, it is not quite suitable to be adopted on nano-networks due to their limited computing capabilities [15].

B. MACHINE LEARNING ALGORITHMS

In general, there are various standard ML algorithms, which can be applied to almost any data problem. Each algorithm has its benefits and limitations. In this paper, we are trying to answer a simple question: which type of machine learning model should be generally used in the analysis of electromagnetic nano-network traffic that crosses the micro/nano-gateway. So, we discuss the most common and the most popular algorithms that were previously used in the literature for the analysis and classification of wireless network traffic. They are decision tree classifier, support vector machines, K-nearest neighbors, random forest and naïve bayes. Each learning algorithm has certain hyper-parameters or values, which can be tuned to get better results by experimentally finding the best combination of values per hyper-parameter.

1) DECISION TREES

A decision tree is an acyclic graph used to make decisions. These graphs are composed of leaves/branches and nodes, where inner nodes correspond to a specific feature of the input vector and leaves are the outcome. If the value of the feature is below a certain threshold, the left branch is followed; otherwise, the right branch is followed. At the end of a leaf node, the class to which the example belongs is made. Classification trees are very popular classification algorithms due to their simplicity, as they can be easily converted into a rule-based classification system. Moreover, they can be graphically represented. The training follows a top-down greedy algorithm that works by iteratively splitting the nodes, using normally an information gain-based metric as an optimization criterion. DTC is mostly used for classification problems. Surprisingly, it works for both categorical and continuous-dependent variables [16], [19].

2) SUPPORT VECTOR MACHINES (SVM)

SVMs are non-probabilistic binary classifiers. They are considered one of the most powerful supervised classification algorithms, as they work by representing each feature vector in a multidimensional space and trying to find a linear separation for the classes. In some cases, a linear separation of the space is not possible and cannot provide a solution, hence the kernel trick is used by increasing the dimensionality of space, making an easier separation in a much higher dimensional space [16], [19].

3) K-NEAREST NEIGHBORS (K-NN)

The K-NN algorithm is a non-parametric approach used for either classification or regression. In both scenarios, the output consists of the K-closest training examples in the feature space. In the K-NN classification, the output is a class membership. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its K-nearest neighbors [16], [19].

4) RANDOM FOREST (RF)

RF is an ensemble learning algorithm that consists of the aggregation of a large number of decision trees, each one based on a different part of the training set. They are randomly selected. These instances are called bootstrapped samples, and the outcome is generally decided by majority voting among all the boot-strapped samples. RF is one of the most widely used ensemble learning algorithms, and it provides a reduction of variance compared to a single decision tree [16], [19].

5) NAÏVE BAYES (NB)

NB is a simple classifier algorithm based on Bayesian statistics. It is popularly used because of its efficiency in multiple scenarios, especially in high-dimensional datasets. Its functionality uses maximum likelihood estimation by assuming that data features are mutually independent, which is not true in most cases. This assumption provides an easy calculation for the class-conditional probabilities [16], [19].

C. PERFORMANCE METRICS

In general, it is difficult to measure the quality of a given model without quantifying its performance in the training and testing phases. This is typically achieved by using some type of performance metric, whether it is through calculating some type of error, the goodness of the model fit, or some other useful measurement. When performing classification predictions, there are four types of outcomes that could occur. These outcomes control the performance metrics and form a matrix called the confusion matrix, which is used to evaluate the quality of the output of a classifier. The diagonal elements represent the number of points for which the predicted label is equal to the true label, while off-diagonal elements are those that are mislabeled by the classifier. The higher the diagonal values of the confusion matrix, the better the indicator for many correct predictions. These outcomes are as follows: i) true positives occur when the model predicts an observation belongs to a class and it truly does belong to that class, ii) true negatives occur when the model predicts an observation does not belong to a class and it truly does not belong to that class, iii) false positives occur when the model predicts an observation belongs to a class when in reality it does not and iv) false negatives occur when the model predicts an observation does not belong to a class when in fact it does [19].

The three main metrics used to evaluate a classification model are accuracy, precision, and recall. The accuracy is defined as the percentage of correct predictions for the test data. Precision is defined as the fraction of relevant examples (true positives) among all of the examples that were predicted to belong in a certain class, while recall is defined as the fraction of examples that were predicted to belong to a class with respect to all of the examples that truly belong in the

class. F_{score} is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall. It has a β factor, which indicates how much more important the recall is than the precision or vice-versa. For example, if the recall is considered to be twice as important as precision, then β equals two, while if both recall and precision have equal importance, then β equals one, which is the standard $F1_{score}$ or usually known as F_{score} [16], [19].

D. TRAFFIC CLASSIFICATION AND PREDICTION

Traffic classification is an important process for telecommunication networks to observe a wide range of operations, measurements and management activities [19]. In nano-networks, traffic classification can be useful for performance monitoring, resource provisioning, traffic prioritization, self-configuration devices, network management, QoS and security by identifying unknown traffic or detecting anomaly behavior to maintain adequate nano-communication.

Generally, traffic classification techniques can be categorized based on port number, payload, or host behavior, but these techniques are considered traditional approaches, and they are not widely applicable. As port-based techniques are unreliable due to the use of dynamic port assignment, tunneling and port number changes to avoid firewalls. While payload-based techniques are computationally intensive and complicated due to encryption. Similarly, host-based traffic classification techniques are highly susceptible to routing asymmetries. In contrast to these legacy approaches, machine learning is the most widespread technique used for traffic classification recently. As supervised and unsupervised, ML has been successfully employed for traffic classification, showing high accuracy and applicability to large datasets [19].

Different studies in the literature provide analysis of traffic classification based on machine learning and investigate QoS support using different datasets. The authors in [20] proposed a classification method to discover traffic characteristics and dynamically assign service classes to IP packets. They applied semi-supervised machine learning techniques, considering packet characteristics such as the unbalanced traffic distribution between classes. While the authors in [21] used SVM to classify the network traffic in campus backbone networks. In their study, SVM achieved reliable and accurate results using biased and unbiased test samples. However, they only analyzed one ML algorithm, neglecting other algorithms.

The authors in [22] compared six supervised machine learning algorithms for traffic classification in the backbone network. Their datasets contain Internet Service Provider (ISP) data traffic, and they used principal component analysis for feature extraction and analyzed its influence on the classification results. While the authors in [23] compared six supervised learning algorithms for traffic classification. Their experiments were conducted using two feature selection methods and five traffic classes. While the authors in [24] proposed a traffic classification technique for smart cities using four supervised machine learning algorithms on datasets containing campus data traffic aiming to improve the QoS in smart city networks.

In electromagnetic nano-networks, traffic prediction and classification are active research fields. Exploring the traffic features enables intelligent resource management and provisioning. Different applications of nano-networks and the heterogeneity of their nano-devices provide a high-level diversity in the generated traffic types. This diversity results in growing randomness in traffic and makes it increasingly difficult to be accurately predicted and classified. Therefore, the high-accuracy prediction of wireless traffic is important to allow automatic network resource allocation with real-time demands [25].

E. MACHINE LEARNING IN IoT AND IoNT DOMAINS

Rapid developments in hardware, software and communication technologies have facilitated the emergence of Internet-connected sensor devices, which provide observations and data measurements from the physical world. In this manner, the IoT and IoNT systems will be able to access raw data from different resources over the network and analyze this information to extract knowledge [17].

Since IoT and IoNT are among the most significant sources of new data, data science provides a considerable contribution to making their applications more intelligent. Data science is a combination of different scientific fields that use data mining, machine learning and other techniques to find patterns and new insights from data by applying analytics methods to particular areas, involving defining data types such as volume, variety and velocity; data models such as neural networks, classification and clustering methods. Also, applying efficient algorithms that match the data characteristics [17].

Due to the growth in the development of smart objects, IoT and IoNT have enriched almost all aspects of our daily lives and are continuously doing so with a diverse range of novel, innovative and intelligent applications. Recently, there has been a surge in the application of ML-based techniques for various IoT and IoNT applications. These applications include smart healthcare, smart cities, smart agriculture and military services [26]. ML techniques have been adopted to solve several challenges in IoT and IoNT. When a huge number of sensors/nodes are randomly deployed in a nano-network, vital issues need to be considered while designing the network such as topology changes, communication link failures, memory constraints of nano-sensors, computational capabilities, decentralized management, localization, clustering, data aggregation, query processing, real-time routing, data integrity and fault detection. Therefore, sophisticated ML techniques make it possible to solve these issues by allowing the nano-network to learn from the previous example scenarios, adapt itself to the dynamic environment, reduce the complexity and extend the network lifetime by saving the energy of nano-nodes [27].

Nano-technology in medicine has a recent notable impact on hospital systems, where the placement of sophisticated nano-scale devices inside the human body can provide remote health monitoring and telemedicine services [10]. One of the emergent applications of telehealth monitoring is the Internet of Medical Things (IoMT), which integrates low-powered nodes to collect, monitor, process and transfer bio-signals providing interconnection of healthcare devices and sensors to the Internet to enable a new class of applications relying on medical data processing and storage [28], [29]. By implanting these nano-scale devices inside the human body, a network called Intrabody Nano-network (IBNN) appeared. This network has tremendous potential for revolutionizing the healthcare structure, as it has a wide range of medical applications by collecting and monitoring physiological parameters for diagnostic and treatment purposes [10], [30].

Over the past years, AI and ML have gained popularity as promising techniques to overcome the complexity, scalability and decision accuracy challenges of resource constraint sensor networks, which can resolve complex problems and provide simple and understandable solutions for dynamic environments and distributed systems [13]. Authors in [13] integrated the characteristics of artificial intelligence to boost the computational intelligence of IBNN. They presented a study motivated by the aforementioned constraints of IBNN and discussed a novel data aggregation scheme for IBNN.

On the other hand, machine learning and deep learning algorithms that are applied in the healthcare domain allow health professionals to monitor, diagnose, focus and highlight the region of the problem and propose the required and accurate solution in the shortest duration possible [28]. In healthcare solutions, the integration of ML algorithms is imperative. As smart-connected wearables collect data on an unprecedented scale, ML techniques are used to perform complex analysis, intelligent judgments and creative problem-solving on the big data generated from these smart wearables/sensors [26]. Moreover, with the aid of ML algorithms, valuable information can be extracted from the acquired data and draw useful inferences [31].

ML models can be trained and tested using large volumes of data, then through inductive conclusion, they can assist medical people in assessing risk and designing the appropriate treatment, thus improving efficiency and reducing errors compared to manual efforts. Also, monitoring, managing and analyzing medical reports will be easier as ML models can process large sets of biological data and detect specific patterns and mutations involved in various diseases. As a result, health monitoring services and consulting can be provided digitally to a certain limit [31].

Authors in [32] collected a large volume of big data with different types such as image, text and categorical data through IoT devices and stored it in a secure cloud environment that can be accessible by healthcare applications. Then, they applied a new machine learning algorithm for proceeding the learning process which maps the data into two classes; normal class and disease affected. While the authors in [33] proposed a traffic classification method for smart cities in IoT networks to remove the manual selection of network traffic features. Their method relied on DL.

Authors in [34] studied the ability to track the location and monitor the health of the soldiers in real-time to know who is lost and getting injured on the battlefield to reduce searching time and minimize rescue operation efforts of the army control unit. They proposed a system with a control unit using the Global Positioning System (GPS) and Wireless Body Area Sensor Networks (WBASNs) that enables tracking the location, monitoring the health of soldiers and collecting their body measurements such as temperature, heartbeat, etc. In their model, the data comes from sensors while a GPS receiver is transmitting wirelessly using a ZigBee module. Their collected data is uploaded to the cloud for further data analysis and predictions are achieved by the K-Means Clustering algorithm.

Recent advances in artificial intelligence and machine learning promise to address emerging communication systems, as they change the conventional operation and structure of legacy networks in many aspects, such as design, infrastructure management, cost reduction and performance improvement [35]. ML techniques have been recently involved with nano-scale communication to optimize and enhance its performance. Generated datasets from nano-communication systems are too vast and complex to parse without computational assistance, so ML can play this vital role effectively by analyzing and extracting new insights. The employment of ML with nano-communication can be classified into multiple distinctive categories such as structure and material design, signal processing and biomedicine applications [14].

In electromagnetic nano-communication architecture [8], Micro/nano-gateways are important devices to ensure the interoperability between different communication protocols and communication domains. They are responsible for encapsulation/decapsulation of packets, translation of addresses, enabling data storage, providing traffic prioritization and forwarding packets to allow proper communication techniques between nano-devices themselves or the Internet [36]. As a result, these gateways process different data formats and traffic patterns from various communication domains. Accordingly, ML-enabled micro/nano-gateways will provide massive communication facilities to the nano-network taking into consideration the limited computing resources and capabilities of nano-devices due to their nano-scale size [37]. Although big data analytics and machine learning have been extensively researched, there is a lack of studies that exclusively focus on the evolution of ML-based techniques for big data analysis in the IoNT domains [26].

F. SUMMARY

Data-driven nano-network can be optimized by enabling automatic analysis of the traffic volume and data measurements generated from nano-sensors/devices. Because of the complexity of nano-networks and their limited computing

resources, special attention has been introduced to ML methodologies to analyze and extract new insights supporting nano-scale communications and their applications. Because nano-sensors generate data in various types with different packet formats towards the micro/nano-gateways, it is important to develop a machine learning model that can handle this heterogeneity and provide a proper traffic classification scheme. This is a very important procedure that unleashes the intelligence of the micro/nano-gateway and adds remarkable advantages for nano-network communication. As illustrated in Table 1, there are multiple works in the literature discussing traffic classification using machine learning on different datasets generated from various networks, such as IP backbone networks, IoT and WBASN supporting various applications. None of the reviewed papers discuss using machine learning techniques to classify nano-network traffic or support IoNT applications, which is the main motivation behind the proposed work of this paper.

III. TRAFFIC GENERATION AND DATASET FEATURES

In this section, we propose the implementation of the electromagnetic nano-network traffic generator and provide a brief illustration of the generated dataset and its features.

A. THE PROPOSED TRAFFIC GENERATOR

We developed a packet generator to generate nano-network traffic combining TCP and UDP traditional network traffic to simulate the traffic received by a micro/nano-gateway. The gateway receives both types of traffic from the nano-scale communication side and the traditional network side. It is worth noting that in AI/ML there is a heavy reliance on a real dataset, as it has more certainty when compared to synthetic data. However, the main reason why synthetic data is used through a generation model is that synthetic data may be more cost-effective and efficient than collecting real-world data in some cases. At present, real-world nano-network traffic is costly to collect due to the lack of availability and the difficulty of having fully operational nano-sensors work in functional nano-network. Accordingly, at the time of simulating this model, there was no access to certain datasets representing the real nano-network traffic and its packet format. So, to overcome this challenge, and as the quality of synthetic data is highly dependent on the quality of the model that creates it, the design of the proposed nano-network traffic generator has been motivated by an existing nano-network simulator, i.e., NanoSim that is discussed in [38]. Hence, a synthetic dataset has been created and used in the simulation. This synthetic data can mimic many properties of authentic data from the NanoSim simulator, especially the packet header format of the nano-sensor payload.

NanoSim simulator is an event-based NS3 simulator used for modeling electromagnetic nano-network communication and its protocol stack. In this simulator, the message processing unit creates random packets generated periodically with a packet size adjusted by the end-user. The network layer adds a network header independent of the routing technique, while the MAC layer follows a certain strategy that has no kind of control, hence the packet is transmitted from the network layer to the physical interface of the nano-router directly based on a technique called transparent-MAC [38].

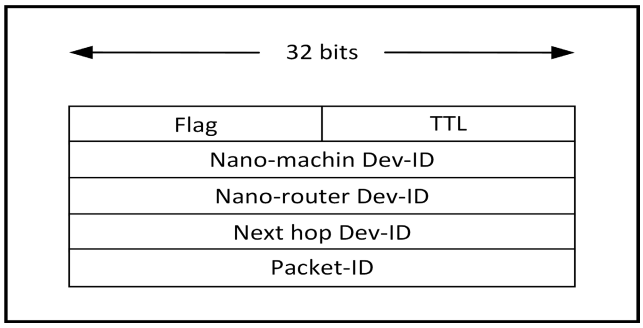

FIGURE 1. Nano-network message header format [38].

In this simulator, the generated message from the nano-device has a fixed header structure consisting of five fields. They are the flag field, the source Dev-ID, the destination Dev-ID, the packet-ID and the Time To Live (TTL). Figure 1 illustrates the nano-network header packet blueprint, where the flag field has been added to indicate if the packet is forwarded to the micro/nano-gateway or a nano-router. Source and destination Dev-IDs are set to the Dev-ID of the local nano-machine and the receiver, respectively. Dev-ID is a unique identifier, and it is assigned to nano-devices, e.g., nano-gateway, nano-router and nano-machine. The structure of the header added by the network layer, as well as the value and the length of each field, can be modified depending on the use case. The packet ID is assigned by the message process unit sequentially, whereas the TTL is set to a value between (100-1000), and it is decreased by one after each hop. The high value of TTL is considered under the assumption that a message on a wireless nano-network can reach its destination after traversing hundreds of nano-devices [38], [39]. Accordingly, the generated bytes stream from a nano-router includes the nano-network packet header associated with the encapsulated payload, then this stream is transmitted through the physical interface of the nano-router towards the micro/nano-gateway.

Our proposed traffic generator generates data flow, which consists of multiple random packets composed of four types. They are TCP packets, UDP packets and nano-network packets, which can be classified into two types, nano-to-nano-communication packets (NN0) and nano-to-Internet-communication packets (NN1). The first type of nano-packet represents the communication scenario between nano-routers, i.e., when a nano-router transmits a packet to another nano-router, while the second type of nano-packets represents the communication scenario with the Internet, i.e., when a nano-router transmits a packet to the Internet or the cloud. Figure 2 illustrates the flowchart of the methodology used by the proposed traffic generator algorithm, which provides the synthetic dataset used in our analysis.

TABLE 1. Some of the proposed work in the literature discussing traffic classification using machine learning.

Traffic classification	Machine learning type	Infrastructure	Application
Proposed in [20]	Semi-supervised learning	IP backbone network	Improve QOS of the ISP network
Proposed in [21]	Supervised learning	IP backbone network	Improve QOS of the campus network
Proposed in [33]	Deep learning	IoT network	Smart cities network
Proposed in [22]	Supervised learning	IP backbone network	Improve QOS of the ISP network
Proposed in [23]	Supervised learning	IP backbone network	Improve QOS of the ISP network
Proposed in [24]	Supervised learning	IP backbone network	Smart cities network
Proposed in [32]	Supervised learning	IoT network	Healthcare application
Proposed in [34]	Supervised learning	WBASN	Military and healthcare applications

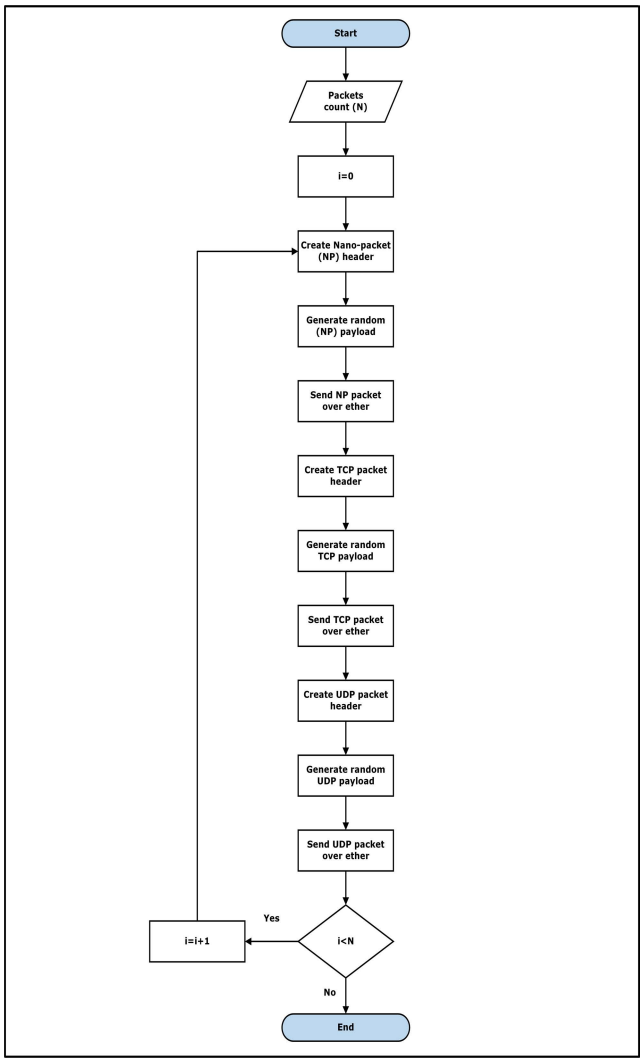

FIGURE 2. Flowchart represents the general structure of the algorithm used by the proposed traffic generator.

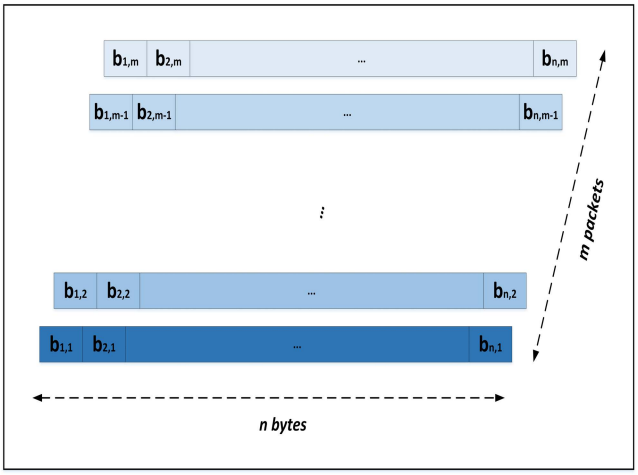

FIGURE 3. Flow representation of the input traffic.

B. DATASET FEATURES EXPLORATION

Input representation of the data is a key element to be considered when building the ML model. We take two types of raw input representations: packets and flows. Figure 3 illustrates the flow representation of the input data (N,m,n), where (N) represents the number of flows, (m) represents the number of packets and (n) represents the number of bytes inside a certain packet. The generated dataset we investigate here is composed of one flow, which consists of 300 packets or sample points. Each sample represents a random packet received by the micro/nano-gateway whether from the nano-network domain or the traditional network domain. This synthetic dataset is used for training and validation tests. Every sample consists of 17 features and one label, which is the target output of the classifier.

Table 2 describes the specific set of 17 features, which are computed for every sample/bin in the proposed dataset. The set includes the number of collected packets and the corresponding fields for each one. There are some features related to nano-network traffic that are characterized by a vector of features containing the source device identification, the sender device identification, the next-hop device identification, the packet identification, besides the flag identification and time to live value. While other features belong to the traditional network traffic, such as source and destination MAC addresses, source and destination IP addresses, transport protocol numbers, and finally, source and destination port numbers. Moreover, other input values are calculated from these fields based on the observation and characteristics of each received packet, such as header size, payload size and packet size.

Table 3 describes the label for every packet, which is the desired output. They are classified into four categories. Two categories are related to traditional network traffic, i.e., TCP and UDP packets, and the other two packets are related to nano-network traffic, i.e., nano-to-nano-communication packet and nano-to-Internet-communication packet.

TABLE 2. Input features for micro/nano-gateway traffic classification and prediction.

Field	Feature description
flag_id	Flag identification
ttl	Time to live
source_dev_id	Source nano-device identification
sender_dev_id	Sender nano-device identification
next_hop_dev_id	Next hop nano-device identification
packet_id	Packet identification
source_mac	Source MAC address
destination_mac	Destination MAC address
source_IP	Source IP address
destination_IP	Destination IP address
transport_protocol	IP transport protocol number
source_port	Source port number
destination_port	Destination port number
payload	Message
payload_size	Message size
header_size	Header size
packet_size	Packet size

TABLE 3. Output labels for the micro/nano-gateway traffic classification and prediction.

Field	Label description
nn0	Nano-to-nano-communication packet
nn1	Nano-to-Internet-communication packet
tcp	TCP packet
udp	UDP packet

Feature selection primarily focuses on removing non-informative features that are not useful or relevant for the model in the classification problem. Intuition suggests that the ttl, the packet_id and the payload are non-informative features for the classification. While the payload_size, the header_size, the packet_size, the source_dev_id, the sender_dev_id, the next_hop_dev_id, the source_mac, the destination_mac, the source_IP and the destination_IP are the main features needed to classify the packet, whether it is a nano-domain packet or a traditional network packet. Furthermore, inside the nano-domain classification, the flag_id plays a vital role in deciding whether the packet is a nano-to-nano-communication packet (NN0) or a nano-to-Internet-communication packet (NN1). Whereas features such as the transport_protocol, the source_port, the destination_port, the payload_size, the header_size and the packet_size are very important to classify if the packet is TCP or UDP.

Data preprocessing is applied to provide cleaned, formatted and restructured data before using it with the ML algorithms. As learning algorithms expect the input to be numeric, some adjustments of certain non-numeric and categorical features need to be converted. One popular way to convert categorical variables is by using the one-hot encoding scheme. This preprocessing step provides predictive power for all learning algorithms. After categorical variables have been converted into numerical features, all numerical features have been normalized. The dataset (both features and their labels) will be split into training and testing sets. In our experiment, 75% of the data is used for training and 25% for testing. Typically, the data is also shuffled into a random order when creating the training and testing subsets to remove any bias in the ordering of the dataset.

IV. PERFORMANCE EVALUATION OF LEARNING MODELS

In this section, we develop the tools and techniques necessary for a model to make the micro/nano-gateway able to classify the nano-network traffic properly. The model will classify traditional traffic from nano-network traffic, where traditional traffic can be classified into TCP or UDP packets, while nano-network traffic can be classified into nano-to-nano-communication domain packets or nano-to-Internet communication domain packets. We investigate five different algorithms, all of which are supervised learners and determine which is best at modeling the data. These models are decision tree classifier, support vector machines, the K-nearest neighbors, random forest and naïve bayes. We start by evaluating the performance achieved by the five learning models and then present an optimization technique for all approaches by tuning their corresponding hyper-parameters. After that, a full comparison between all approaches has been presented. For the sake of training and testing, we consider 10-fold cross-validation in all the results presented in this section. Parameters on each optimized algorithm are calibrated based on the best-performance grid search test.

In this simulation, we answer the question of which type of machine learning model is optimal for the analysis and classification of micro/nano-gateways traffic. This is achieved by calculating the accuracy, F_{score} and confusion matrix for each model to quantify its performance. These metrics are very useful in the analysis, as they describe how good the model is at making predictions and classifications. Moreover, we draw the learning curves for both training and cross-validation sets to diagnose and visualize the performance during the learning phase, i.e., to figure out how much a machine learning model benefits from adding more training data or samples, and whether the estimator suffers from a high bias (underfitting) or high variance (overfitting). We achieve these objectives by implementing the following steps for each classifier:

- Calculate the accuracy and F_{score} for the unoptimized model and check its learning curves.
- Tune the corresponding hyper-parameters of the model.
- Calculate the accuracy and F_{score} for the optimized model and check its learning curves.
- Calculate the normalized confusion matrix for the optimized model.

A. DECISION TREE CLASSIFIER

By fitting the decision tree classifier model, Figure 4 visualizes the learning curves of the unoptimized model for both training and cross-validation sets as the size of the training set increases. It is obvious that the model goes to overfitting after almost 50 training points, showing 100% accuracy and F_{score} for both training and testing sets. Overfitting is a problem that a model can exhibit, where the

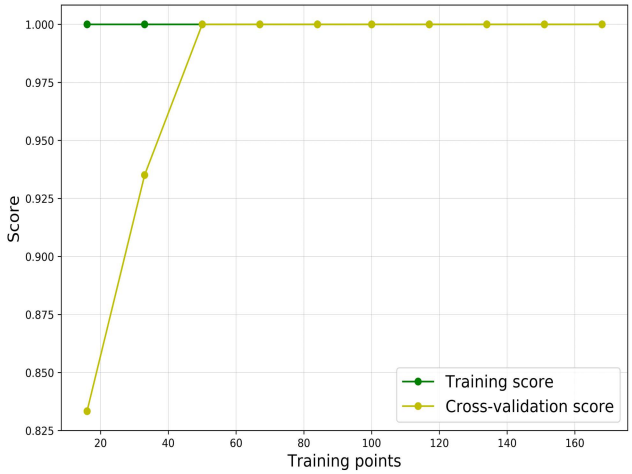

FIGURE 4. Learning curves for unoptimized DTC model.

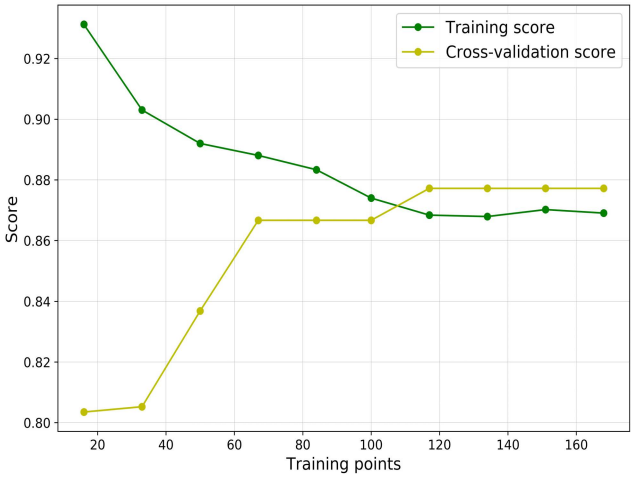

FIGURE 5. Learning curves for optimized DTC model.

model predicts very well the training data but poorly predicts the testing set. Some reasons can lead to overfitting, such as the complexity of the model for the data, and the existence of many features with a small number of training examples.

In order to avoid overfitting, we manipulate the hyper-parameters of the model using the grid search optimization method to enhance its accuracy and F_{score}. We use maximum depth, minimum samples per leaf and minimum samples per split as tuning parameters. Figure 5 shows the learning curves of the optimized model. By increasing the training model points/samples, the training score and cross-validation score converge to almost a score that equals 87% at reaching 110 training samples. With more increase in training points, the DTC model will suffer from overfitting.

After calculating the performance metrics for the optimized model, the overall accuracy equals 87.11%, while F_{score} equals 82.29% for the training set. Meanwhile, for the testing set, it shows 77.33% and 70.94% for the accuracy and F_{score} respectively. Moreover, the optimized values for the hyper-parameters are; maximum depth equals 2, the minimum samples per leaf equal 1 and the minimum samples per split equal 2.

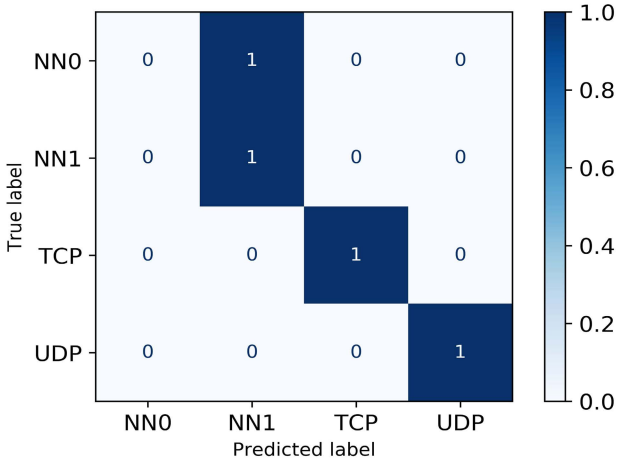

FIGURE 6. Normalized confusion matrix for optimized DTC model.

To dig deep into the out-performance of the model, Figure 6 depicts the corresponding normalized confusion matrix obtained with the DTC model. This shows that the model perfectly classifies TCP, UDP and NN1 packets. While it fails to recognize any NN0 packet, as it predicts all of them to be NN1 packets. This means that the model can distinguish traditional traffic from nano-network traffic correctly, but it cannot determine the exact type of packet within nano-domain communication.

B. SUPPORT VECTOR MACHINES

By fitting the support vector machine model, Figure 7 illustrates the learning curves for the unoptimized model. As the training points increase, both the training and cross-validation score will increase until it converges between (85%-90%) at almost 150 training points. With more increase in training points, the SVM model will suffer from overfitting. This unoptimized model shows that the overall accuracy equals 87.11% and F_{score} equals 82.29% for the training set, while for the testing set it shows 77.33% and 70.94% for the accuracy and F_{score} respectively.

By tuning the hyper-parameters of the SVM model using the grid search optimization method, we obtain better values for the performance metrics. We use the C-parameter, kernel and degree as hyper-parameters for the model. Figure 8 shows the learning curves of the optimized model. As the training points increase, the training score decreases and the cross-validation score increases without overlapping for all training samples. After calculating the performance metrics of the optimized model, the overall accuracy percentage equals 91.11% and F_{score} equals 90.11% for the training set. While for the testing set, it shows 80% and 78.55% for the accuracy and F_{score} respectively. Moreover, the optimized values for the hyper-parameters are; C-parameter equals 20, the kernel is poly and the degree equals 2.

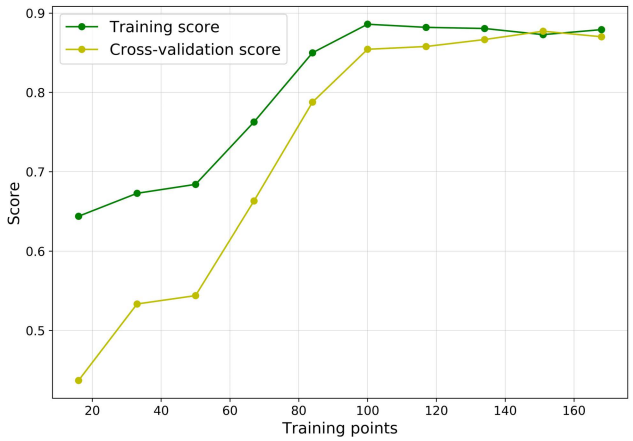

FIGURE 7. Learning curves for unoptimized SVM model.

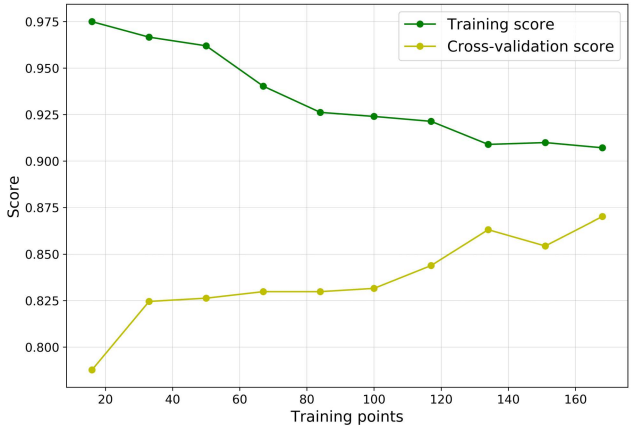

FIGURE 8. Learning curves for optimized SVM model.

Figure 9 shows the normalized confusion matrix for the optimized SVM model. This shows that the model perfectly classifies all traditional network traffic, i.e., TCP and UDP packets. While it provides correct predictions for 82% of the received NN1 packets, and only 24% of the received NN0 packets. This means that the model can distinguish traditional traffic from nano-network traffic correctly, however it gives some faulty predictions for the packet class of nano-domain communication. It is obvious that the model is biased more towards the NN1 packet class, and it predicts 76% of the NN0 class to be NN1.

C. THE K-NEAREST NEIGHBORS

By fitting the K-nearest neighbors model, Figure 10 shows the learning curves for the unoptimized model. It shows that as training points increase, the score of both the training and cross-validation sets increases without overlapping for all training points. The unoptimized mode shows accuracy equals 89.33% and F_{score} 88.77% in the training set, while for the testing set it shows 82.66% and 82.06% for accuracy and F_{score} respectively. After manipulating the number of neighbors as a hyper-parameter for the KNN model, the performance metrics of the model have been enhanced.

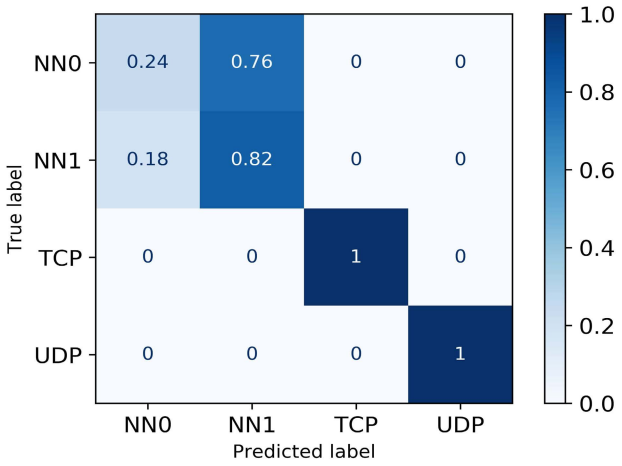

FIGURE 9. Normalized confusion matrix for optimized SVM model.

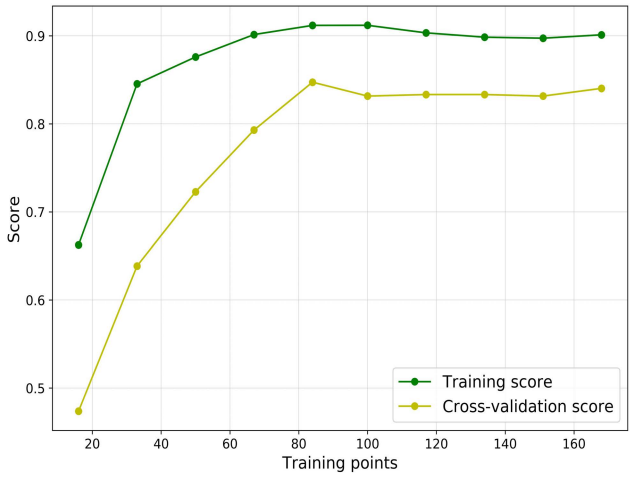

FIGURE 10. Learning curves for unoptimized KNN model.

Figure 11 shows the learning curves of the optimized KNN model, the model can avoid overfitting and underfitting for all training points with overall accuracy equals 91.11% and F_{score} equals 90.88% in the training set, while for the testing set it shows 82.66% and 82.06% for accuracy and F_{score} respectively. Besides, the optimized number of neighbors equals 3.

Figure 12 shows the normalized confusion matrix for the optimized KNN model. It shows that the model classifies perfectly TCP and UDP packets. While it provides correct predictions for 82% of the received NN1 packets, and 35% of the received NN0 packets. This means that the model can distinguish traditional traffic from nano-network traffic correctly, however it gives some faulty predictions for the packet class of nano-domain communication. The model is biased towards the NN1 packet class, but with better performance than the optimized SVM model. Because the KNN model provides correct predictions for 35% of the received NN0 packets, but it fails to classify the rest of them successfully, which are predicted wrongly under the NN1 class.

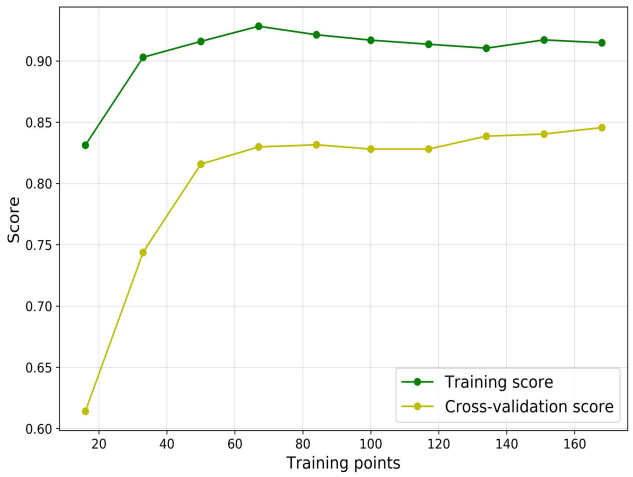

FIGURE 11. Learning curves for the optimized KNN model.

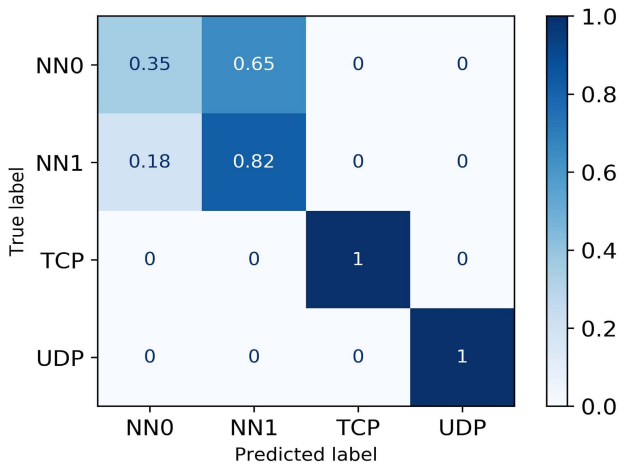

FIGURE 12. Normalized confusion matrix for optimized KNN model.

D. RANDOM FOREST

By fitting the random forest model, Figure 13 visualizes the learning curves for the unoptimized model, which goes to overfitting at 100 training points for both training and cross-validation sets. The model shows 100% overall accuracy and F_{score} for both training testing sets. To avoid overfitting, we manipulate some of the corresponding hyper-parameters of the RF model such as maximum depth, minimum samples per leaf and minimum samples per split.

Figure 14 shows the learning curves for the optimized RF model with overall accuracy equals 87.11% and F_{score} equals 82.29% in the training set, while for the testing set it shows 77.33% and 70.94% for accuracy and F_{score} respectively. Moreover, the optimized values for the hyper-parameters are; maximum depth equals 2, the minimum samples per leaf equal 9 and the minimum samples per split equal 8.

Figure 15 shows the normalized confusion matrix for the optimized RF model. This shows that the model perfectly classifies TCP, UDP and NN1 packets. While it fails to recognize any NN0 packet, as it predicts all of them to be NN1 packets. This means that the model can distinguish traditional traffic from nano-network traffic correctly, but it cannot determine the exact type of packet within the nano-domain communication, which is the same case as the DTC model.

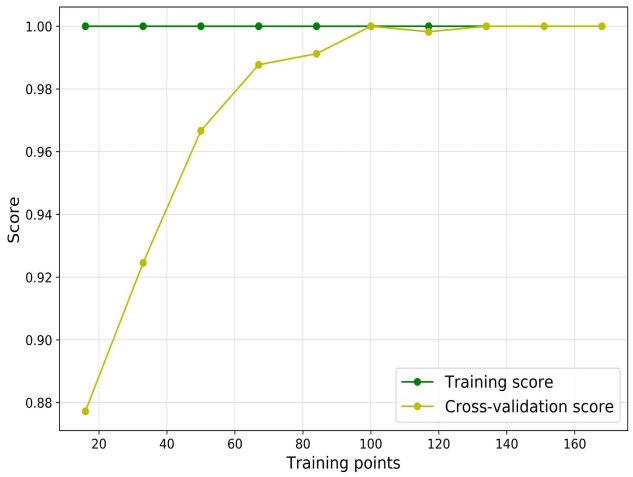

FIGURE 13. Learning curves for unoptimized RF model.

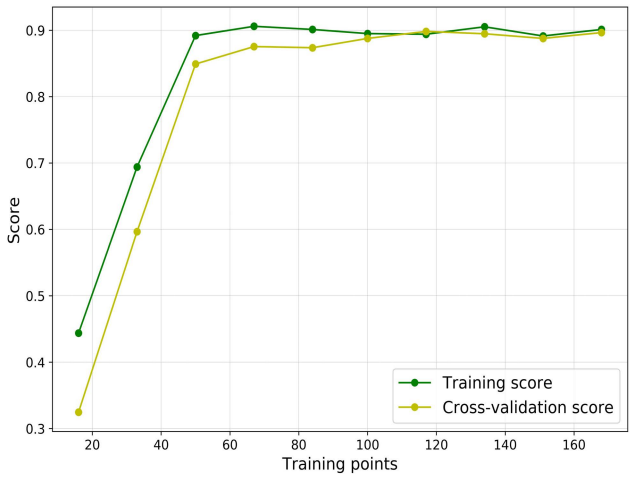

FIGURE 14. Learning curves for optimized RF model.

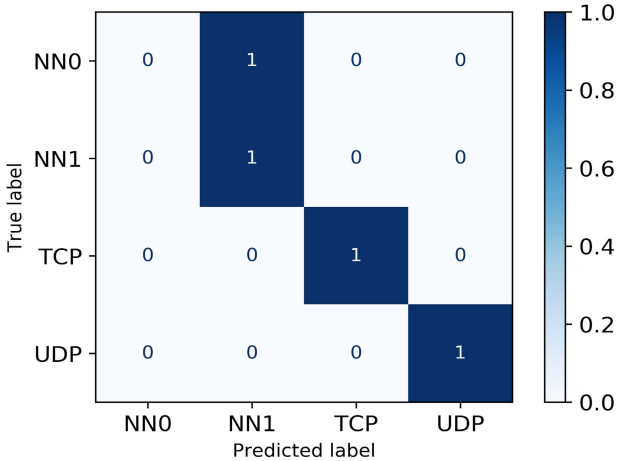

FIGURE 15. Normalized confusion matrix for optimized RF model.

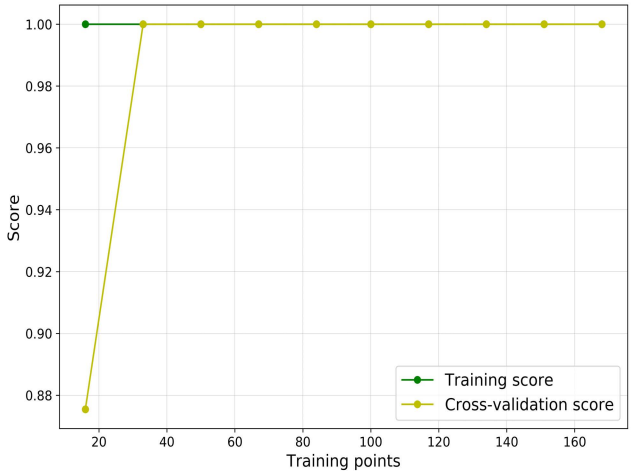

FIGURE 16. Learning curves for unoptimized NB model.

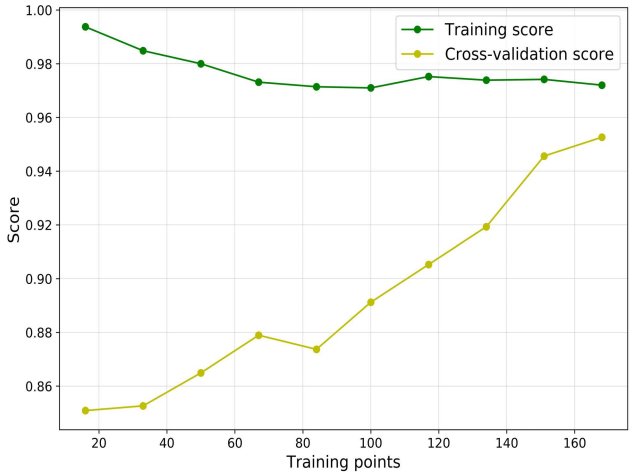

FIGURE 17. Learning curves for optimized NB model.

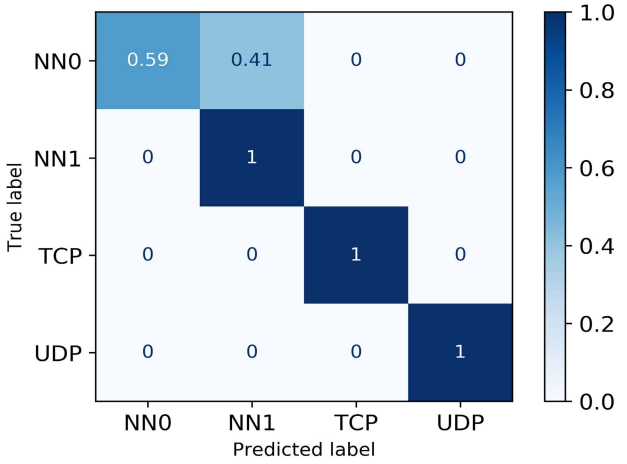

FIGURE 18. Normalized confusion matrix for optimized NB model.

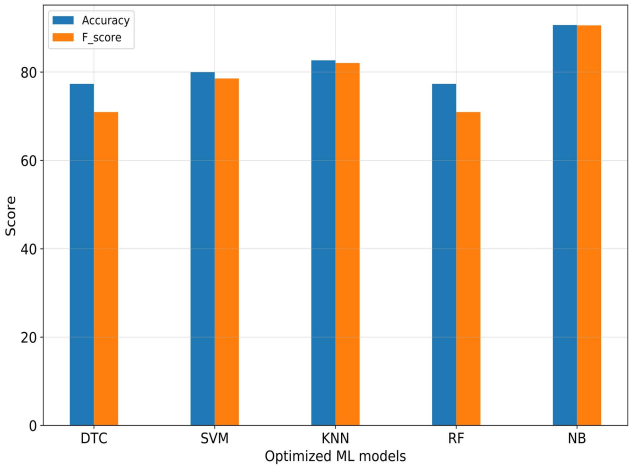

FIGURE 19. Accuracy and F_{score} for testing dataset of the optimized ML models.

E. NAÏVE BAYES

By fitting the naïve bayes model, Figure 16 illustrates the learning curves for the unoptimized model, which goes to overfitting showing 100% accuracy and F_{score} for both training and testing sets after almost 30 training points.

After manipulating the hyper-parameter smoothing of the NB model, Figure 17 shows the learning curves for the optimized NB model. By increasing the training points, the score of the training set is decreased while the score of the cross-validation set is increased for all training points until they converge around 96%. The NB model shows that the overall accuracy equals 97.77% and F_{score} equals 79.73% in the training set, while for the testing set it shows 90.66% and 90.58% for accuracy and F_{score} respectively. Moreover, the optimized values for smoothing hyper-parameter equals 0.0001.

Figure 18 illustrates the normalized confusion matrix for the optimized NB model. This shows that the model perfectly classifies all TCP, UDP and NN1 packets. While it provides correct predictions for 59% of the total received NN0 packets. This means that the model can distinguish traditional traffic from nano-network traffic correctly, moreover it provides quite reasonable predictions for the packet class of the nano-domain communication. The NB model provides better performance compared to the SVM and KNN models, as it can predict 59% of the received NN0 packets, but it fails to classify the rest of them (41%) successfully, which are predicted wrongly under the NN1 class.

Table 4 provides a summarized comparison between accuracy and F_{score} for all optimized and unoptimized five models, while Figure 19 visualizes the accuracy and F_{score} for the testing set of all optimized models. It is obvious that both the optimized NB and KNN models provide a better fit for the micro/nano-gateway traffic classification problem with higher accuracy and F_{score} values from DTC, SVM and RF models. In addition, both models show a good fit for the dataset during the training and cross-validation phases without overfitting or underfitting for all training bins/samples. However, the NB model outperforms the KNN model, as it provides better accuracy, F_{score} and normalized confusion matrix. Hence, the NB model is the best fit for the micro/nano-gateway traffic classification problem.

TABLE 4. Comparison between accuracy and F_{score} for optimized and unoptimized ML models.

Model	Metric	Dataset type	Unoptimized model	Optimized model
DTC	Accuracy (%)	Training	100	87.11
		Testing	100	77.33
	F_{score}(%)	Training	100	82.29
		Testing	100	70.94
SVM	Accuracy (%)	Training	87.11	91.11
		Testing	77.33	80
	F_{score}(%)	Training	82.29	90.11
		Testing	70.94	78.55
KNN	Accuracy (%)	Training	89.33	91.11
		Testing	82.66	82.66
	F_{score}(%)	Training	88.77	90.88
		Testing	82.06	82.06
RF	Accuracy (%)	Training	100	87.11
		Testing	100	77.33
	F_{score}(%)	Training	100	82.29
		Testing	100	70.94
NB	Accuracy (%)	Training	100	97.77
		Testing	100	90.66
	F_{score}(%)	Training	100	97.73
		Testing	100	90.58

V. CONCLUSION AND FUTURE WORK

AI/ML integration with different levels of telecommunication networks enables operators to predict information, adapt to network changes and manage network resources to achieve a high level of performance targets. Empowering nano-networks with AI/ML functionalities will make a dramatic improvement in their operational techniques, as ML plays a significant role in shaping electromagnetic nano-network functionalities in resource management, monitoring and prediction. In this paper, we develop a nano-network traffic generator to generate nano-network packets combined with traditional background traffic, then we employ five supervised ML algorithms to accurately model and classify micro/nano-gateway traffic using the generated synthetic dataset. The main objective is to construct a model that accurately classifies nano-network traffic received by a micro/nano-gateway. The dataset is collected from the developed packet generator, which generates nano-packets representing the nano-network traffic associated with background traffic composed of multiple TCP and UDP packets that represent traditional traffic. We have demonstrated the outstanding performance of the DTC, SVM, KNN, RF and NB algorithms for the analysis of traffic received by micro/nano-gateway from both macro and nano wireless communication domains. Performance is evaluated for all models in terms of global classification accuracy, i.e., correctly classified instances, F_{score}, learning curves and normalized confusion matrix. Both NB and KNN models showed in general better results in terms of accuracy and prediction than DTC, SVM and RF, however the NB model is nominated to be the best ML model that fits the traffic classification problem due to its higher diagonal confusion matrix. Therefore, the NB model represents a very appealing ML model for electromagnetic nano-network traffic analytics as results suggest it to be the most accurate to address this problem.

Our future work will look at more complex models based on ensemble learning methods, which use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the learning algorithms, hence expanding the corresponding ML studies to improve the detection and prediction accuracy with more enhancement in the performance evaluation metrics.

REFERENCES

[1] A. Oukhatar, M. Bakhouya, D. El Ouadghiri, and K. Zine-Dine, "Probabilistic-based broadcasting for EM-based wireless nanosensor networks," in *Proc. 15th Int. Conf. Adv. Mobile Comput. Multimedia (MoMM)*, New York, NY, USA, 2017, pp. 232–236, doi: 10.1145/3151848.3151872.

[2] M. Pierobon, J. M. Jornet, N. Akkari, S. Almasri, and I. F. Akyildiz, "A routing framework for energy harvesting wireless nanosensor networks in the terahertz band," *Wireless Netw.*, vol. 20, no. 5, pp. 1169–1183, Jul. 2014, doi: 10.1007/s11276-013-0665-y.

[3] N. A. Ali, W. Aleyadeh, and M. AbuElkhair, "Internet of Nano-Things network models and medical applications," in *Proc. Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Sep. 2016, pp. 211–215.

[4] U. A. K. Chude-Okonkwo, R. Malekian, B. T. Maharaj, and A. V. Vasilakos, "Molecular communication and nanonetwork for targeted drug delivery: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 3046–3096, 4th Quart., 2017.

[5] T. Iftikhar, H. A. Khattak, Z. Ameer, M. A. Shah, F. F. Qureshi, and M. Z. Shakir, "Human bond communications: Architectures, challenges, and possibilities," *IEEE Commun. Mag.*, vol. 57, no. 2, pp. 19–25, Feb. 2019.

[6] I. F. Akyildiz, F. Brunetti, and C. Blázquez, "Nanonetworks: A new communication paradigm," *Comput. Netw.*, vol. 52, no. 12, pp. 2260–2279, Aug. 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128608001151

[7] X.-W. Yao, Y.-C.-G. Wu, and W. Huang, "Routing techniques in wireless nanonetworks: A survey," *Nano Commun. Netw.*, vol. 21, Sep. 2019, Art. no. 100250. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1878778919300195

[8] A. Galal and X. Hesselbach, "Nano-networks communication architecture: Modeling and functions," *Nano Commun. Netw.*, vol. 17, pp. 45–62, Jul. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1878778918300164

[9] A. Galal and X. Hesselbach, "Probability-based path discovery protocol for electromagnetic nano-networks," *Comput. Netw.*, vol. 174, Jun. 2020, Art. no. 107246. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128619308801

[10] H. Fahim, W. Li, S. Javaid, M. M. S. Fareed, G. Ahmed, and M. K. Khattak, "Fuzzy logic and bio-inspired firefly algorithm based routing scheme in intrabody nanonetworks," *Sensors*, vol. 19, no. 24, p. 5526, Dec. 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/24/5526

[11] A. Galal, X. Hesselbach, W. Tavernier, and D. Colle, "SDN-based gateway architecture for electromagnetic nano-networks," *Comput. Commun.*, vol. 184, pp. 160–173, Feb. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366421004898

[12] C. Mouradian, N. T. Jahromi, and R. H. Glitho, "NFV and SDN-based distributed IoT gateway for large-scale disaster management," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4119–4131, Oct. 2018.

[13] S. Javaid, Z. Wu, H. Fahim, I. B. Mabrouk, M. Al-Hasan, and M. B. Rasheed, "Feedforward neural network-based data aggregation scheme for intrabody area nanonetworks," *IEEE Syst. J.*, early access, Dec. 29, 2020, doi: 10.1109/JSYST.2020.3043827.

[14] A.-A.-A. Boulogeorgos, S. E. Trevlakis, S. A. Tegos, V. K. Papanikolaou, and G. K. Karagiannidis, "Machine learning in nano-scale biomedical engineering," *IEEE Trans. Mol., Biol. Multi-Scale Commun.*, vol. 7, no. 1, pp. 10–39, Mar. 2021.

[15] P. Casas, "Machine learning models for wireless network monitoring and analysis," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2018, pp. 242–247.

[16] A. Burkov, *The Hundred-Page Machine Learning Book*. Quebec City, QC, Canada: Andriy Burkov, 2019.

[17] M. S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, and A. P. Sheth, "Machine learning for Internet of Things data analysis: A survey," *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 161–175, Aug. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S235286481730247X

[18] H. Bolhasani, M. Mohseni, and A. M. Rahmani, "Deep learning applications for IoT in health care: A systematic review," *Informat. Med. Unlocked*, vol. 23, Jan. 2021, Art. no. 100550. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S235291482100040X

[19] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, "A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities," *J. Internet Services Appl.*, vol. 9, no. 1, p. 16, 2018, doi: 10.1186/s13174-018-0087-2.

[20] D. Aureli, A. Cianfrani, A. Diamanti, J. M. S. Vilchez, and S. Secci, "Going beyond diffserv in IP traffic classification," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2020, pp. 1–6, doi: 10.1109/NOMS47738.2020.9110430.

[21] Y. Zhu and Y. Zheng, "Traffic identification and traffic analysis based on support vector machine," *Neural Comput. Appl.*, vol. 32, no. 7, pp. 1903–1911, 2020, doi: 10.1007/s00521-019-04493-2.

[22] Y. Miao, Z. Ruan, L. Pan, J. Zhang, Y. Xiang, and Y. Wang, "Comprehensive analysis of network traffic data," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT)*, Dec. 2016, pp. 423–430.

[23] M. A. P. Perera, G. Tian, C. Fidge, and W. Kelly, "A comparison of supervised machine learning algorithms for classification of communications network traffic," in *Proc. 24th Int. Conf. Neural Inf. Process. (ICONIP)*, in Lecture Notes in Computer Science, vol. 10634, Y. Li, D. Liu, S. Xie, D. Zhao, and E. El-Alfy, Eds. Cham, Switzerland: Springer, 2017, pp. 445–454. [Online]. Available: https://eprints.qut.edu.au/115355/

[24] R. M. AlZoman and M. J. F. Alenazi, "A comparative study of traffic classification techniques for smart city networks," *Sensors*, vol. 21, no. 14, p. 4677, Jul. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/14/4677

[25] J. Wen, M. Sheng, J. Li, and K. Huang, "Assisting intelligent wireless networks with traffic prediction: Exploring and exploiting predictive causality in wireless traffic," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 26–31, Jun. 2020.

[26] W. Li, Y. Chai, F. Khan, S. R. U. Jan, S. Verma, V. G. Menon, Kavita, and X. Li, "A comprehensive survey on machine learning-based big data analytics for IoT-enabled smart healthcare system," *Mobile Netw. Appl.*, vol. 26, no. 1, pp. 234–252, Feb. 2021, doi: 10.1007/s11036-020-01700-6.

[27] S. Messaoud, A. Bradai, S. H. R. Bukhari, P. T. A. Quang, O. B. Ahmed, and M. Atri, "A survey on machine learning in Internet of Things: Algorithms, strategies, and applications," *Internet Things*, vol. 12, Dec. 2020, Art. no. 100314. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660520301451

[28] S. Durga, R. Nag, and E. Daniel, "Survey on machine learning and deep learning algorithms used in Internet of Things (IoT) healthcare," in *Proc. 3rd Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Mar. 2019, pp. 1018–1022.

[29] S. Pirbhulal, N. Pombo, V. Felizardo, N. Garcia, A. H. Sodhro, and S. C. Mukhopadhyay, "Towards machine learning enabled security framework for IoT-based healthcare," in *Proc. 13th Int. Conf. Sens. Technol. (ICST)*, Dec. 2019, pp. 1–6.

[30] H. Fahim, S. Javaid, W. Li, I. B. Mabrouk, M. Al Hasan, and M. B. B. Rasheed, "An efficient routing scheme for intrabody nanonetworks using artificial bee colony algorithm," *IEEE Access*, vol. 8, pp. 98946–98957, 2020.

[31] H. K. Bharadwaj, A. Agarwal, V. Chamola, N. R. Lakkaniga, V. Hassija, M. Guizani, and B. Sikdar, "A review on the role of machine learning in enabling IoT based healthcare applications," *IEEE Access*, vol. 9, pp. 38859–38890, 2021.

[32] P. M. Kumar, S. Lokesh, R. Varatharajan, G. C. Babu, and P. Parthasarathy, "Cloud and IoT based disease prediction and diagnosis system for healthcare using fuzzy neural classifier," *Future Gener. Comput. Syst.*, vol. 86, pp. 527–534, Sep. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18303753

[33] H. Yao, P. Gao, J. Wang, P. Zhang, C. Jiang, and Z. Han, "Capsule network assisted IoT traffic classification mechanism for smart cities," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 7515–7525, Oct. 2019.

[34] A. Gondalia, D. Dixit, S. Parashar, V. Raghava, A. Sengupta, and V. R. Sarobin, "IoT-based healthcare monitoring system for war soldiers using machine learning," *Proc. Comput. Sci.*, vol. 133, pp. 1005–1013, Jan. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918310202

[35] U. Challita, H. Ryden, and H. Tullberg, "When machine learning meets wireless cellular networks: Deployment, challenges, and applications," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 12–18, Jun. 2020.

[36] O. Salman, I. Elhajj, A. Kayssi, and A. Chehab, "Edge computing enabling the Internet of Things," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, Dec. 2015, pp. 603–608.

[37] I. F. Akyildiz, J. M. Jornet, and M. Pierobon, "Nanonetworks: A new frontier in communications," *Commun. ACM*, vol. 54, no. 11, pp. 84–89, Nov. 2011, doi: 10.1145/2018396.2018417.

[38] G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Nano-sim: Simulating electromagnetic-based nanonetworks in the network simulator 3," in *Proc. 6th Int. Conf. Simulation Tools Techn.*, Jul. 2013, pp. 203–210.

[39] G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Simulating wireless nano sensor networks in the ns-3 platform," in *Proc. 27th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2013, pp. 67–74.

AKRAM GALAL received the B.Sc. degree in electronics and communication engineering from the University of Alexandria, Egypt, in 2010, the post graduate diploma in computer networks from the Information Technology Institute (ITI), and the M.Sc. degree in electronics and communication engineering from the Arab Academy for Science, Technology and Maritime Transport, Egypt, in 2017. He is currently pursuing the Ph.D. degree with the Network Engineering Department, Universitat Politècnica de Catalunya (UPC)—Barcelonatech. From 2011 to 2014, he worked as an Enterprise Networks Engineer at TE Data, Egypt. From 2014 to 2017, he worked as a Solution Design Consultant at Tawasul Telecom, Kuwait. In 2017, he joined design, modeling, and evaluation of broadband networks (BAMPLA) research group as a Research and Development Engineer. His research interests include software defined networking, the Internet of Things, fog computing, network function virtualization, and nano-networks. He is a recipient of FI-AGAUR scholarship awarded by the Agència de Gestió d'Ajuts Universitaris i de Recerca (AGAUR) of the Generalitat de Catalunya co-financed by the European Social Fund (ESF).

XAVIER HESSELBACH (Senior Member, IEEE) received the M.S. degree (Hons.) in telecommunications engineering and the Ph.D. degree (Hons.) from the Universitat Politècnica de Catalunya (UPC), in 1994 and 1999, respectively. In 1993, he joined the design, modeling, and evaluation of broadband networks (BAMPLA) group with the Network Engineering Department, UPC, where he is currently an Associate Professor with the Department of Network Engineering (Department Enginyeria Telematica, ENTEL, https://entel.upc.edu/en). He has been involved in more than 20 national and international competitive projects. He is the author of four books and more than 150 national and international publications in conferences and journals. His research interests include networks virtualization, resources management, SDN networks, quality of service, green networking, and nano-networks. In 1994, he received the award from the COIT/AEIT of Spain for the Best Master Thesis on Networks and Telecommunication Services. He has participated in the technical program committees of more than 30 conferences, he has served as the Information Systems and Internet Chair in Infocom 2006, and a guest editor in several journals. He has taken part in European and Spanish research projects, such as the EuroNGI/FGI/NF Network of Excellence, COST293, Mantychore, and All4Green. For more information visit the link (https://futur.upc.edu/XavierHesselbachSerra).