Denial-of-Sleep Attack Detection in NB-IoT Using Deep Learning

Tahani Bani-Yaseen
Department of Electrical Engineering, School of Engineering, Princess Sumaya University for Technology, Amman, Jordan

Ashraf Tahat
Department of Communications Engineering, School of Engineering, Princess Sumaya University for Technology, Amman, Jordan

Kira Kastell
Office of the President, Hamm-Lippstadt University of Applied Sciences, Hamm, Germany

Talal A. Edwan
Department of Computer Engineering, Faculty of Engineering, Al-Ahliyya Amman University, Amman, Jordan

Abstract: With increasing Internet-of-Things (IoT) protocols and connectivity, a growing number of attacks are emerging in the associated networks. This work presents approaches using deep learning (DL) to detect attacks in an IoT environment, particularly in narrowband Internet-of-Things (NB-IoT). By virtue of its low cost, low complexity and limited energy, an NB-IoT device will not likely permit cutting-edge security mechanisms, leaving it vulnerable to, for example, denial-of-sleep (DoSl) attacks. For performance analysis, a NB-IoT network was simulated, using ns-3, to generate a novel dataset to represent an implementation of DoSl attacks. After preprocessing, the dataset was presented to a collection of machine learning (ML) models to evaluate their performance. The considered DL recurrent neural network (RNN) models have proven capable of reliably classifying traffic, with very high accuracy, into either a DoSl attack or a normal record. The performance of a long short-term memory (LSTM) classifier has provided accuracies up to 98.99%, with a detection time of 2.54 x 10^{-5} second/record, surpassing performance of a gated recurrent unit (GRU). RNN DL models have superior performance in terms of accuracy of detecting DoSl attacks in NB-IoT networks, when compared with other ML algorithms, including support vector machine, Gaussian naïve-Bayes, and logistic regression.

Keywords: Deep learning, denial-of-sleep attack (DoSl), Internet-of-Things (IoT), NB-IoT, recurrent neural network (RNN).

http://doi.org/10.18080/jtde.v10n3.532

Introduction

The evolution and materialization of the Internet of Things (IoT) lead to the concurrent deployment of numerous battery-powered and energy-harvesting user equipment (UE) types to achieve the functions of sensing and actuation in target application domains that include smart cities, precision agriculture, telemedicine, and industrial automation. The number of IoT devices could surpass 70 billion in 2025, with 70% of the deployed devices being in the low-power and low-cost categories. It is apparent that communications for these low-power devices is mostly by means of wireless technologies. It is believed that wireless low-power and low-cost technologies will have their eminence in connecting billions of deployed IoT devices.

Low power wide area networks (LPWAN) have affected a contemporary direction in the IoT ecosystem through furnishing cost-effective connectivity to distributed low-power devices over a relatively broad geographical area (few tens of kilometres), while maintaining a battery life of up to ten years. LPWAN technologies are readily appropriate for the special requirements of machine-to-machine (M2M) and IoT systems. Despite the fact that these technologies are predominantly at the initial stages of commercial implementations, nevertheless they seem to be up-and-coming. A generic LPWAN network configuration consists of an LPWAN sensing or actuation device, a gateway, and a distant application server on the Internet.

Narrowband Internet-of-Things (NB-IoT) is a relatively new cellular LPWAN technology that has been proposed in the 3rd generation partnership project (3GPP) Release 13 for providing wide-area coverage within the field of IoT (Popli, Jha & Jain, 2018). It possesses the previously described advantages of LPWAN networks, in addition to the special feature of flexible wide coverage, using a small frequency bandwidth of as little as approximately 180 kHz on existing cellular technologies (Wang *et al.*, 2017; Tahat *et al.*, 2020). They have vital superiority for nationwide integration with sustained seamless coverage of high capacity.

However, IoT systems in general are confronted with risks and vulnerabilities that are proportionate to the broad scope of IoT applications in multiple vertical industries. They may endure assaults and attacks against physical interfaces or remote communications, in addition to traditional attacks that can be launched against user interfaces, accounts, authentication, and internal communications. Among the various types of attacks that NB-IoT networks can be exposed to are denial-of-service and denial-of-sleep (DoSl) attacks, which are of vital importance, as they not only decrease the performance and efficiency of networks, but they will also shorten the expected overall lifetime of deployed UEs. The expected lifetime of a UE is assumed to be at least 10 years in worst case scenarios, according to requirements of the 3GPP (TR-45.820, 2015).

http://doi.org/10.18080/jtde.v10n3.532

By virtue of its low-cost, low-complexity and limited energy, an NB-IoT device design will not likely permit cutting-edge and impeccable security mechanisms (Chen *et al.*, 2017; TR-45.820, 2015). Hence, this drawback could likely simplify the process of security vulnerability exploitation. Considering the massive numbers of NB-IoT terminals or UEs, any minor vulnerability is potentially capable of inducing critical repercussions to network security. If we envisage a network deployment scheme with NB-IoT terminals which are using a live cellular core network, the device equipment has the capability to infect elements of the mobile core network, including the home subscriber server, the mobility management entity, and supplementary UEs, so as to affect communications of mobile subscribers, resulting in refusal of UE access to the network.

Within this framework, a crucial dilemma is that of denial-of-sleep (DoSl) attacks, as they permanently or transiently dispossess battery-powered or energy-constrained NB-IoT UEs of commencing sleep or energy-saving modes, consequently depleting their stored charge. Alternatively, a viable DoSl attack leads to an extended outage of the compromised NB-IoT UEs. Furthermore, to restore operation of these battery-drained UEs, the tedious task of replacing their batteries needs to be conducted, especially if a battery-powered device is installed at an unattainable setting. Despite the fact that, over the past few years, researchers have proposed ample protection methods against DoSl attacks, the majority of current IoT protocols, including NB-IoT, are not equipped at all with mechanisms to defend against DoSl attacks. Nevertheless, while admitting that there exist abundant DoSl defences, powerful and efficient defences against particular types of DoSl attacks, and for specific connectivity technologies, such as the NB-IoT standard, still need to be developed.

In this paper, we present a deep learning (DL) based approach to defend against DoSl in a NB-IoT network to prevent and protect against the described issues and ramifications. This is accomplished through detecting malicious traffic packets related to an DoSl attack in a model NB-IoT network. To this end, a dataset was generated and subsequently analyzed to train a collection of designated machine learning (ML) algorithms for this goal, including RNN models. This is accomplished by the construction and simulation of an NB-IoT network to be attacked by the *HELLO flood* DoSl attack. The generated traffic was extracted, visualized and pre-processed. All network simulation codes were implemented on the network simulator (ns-3) software suite and written using the C++ programming language. The ML RNN models were implemented using Python on the Google Colaboratory (Colab) platform (Google Colaboratory (Colab), 2021; Bisong, 2019) for training and testing of all samples of the dataset to obtain reliable, consistent, and accurate results for verification. Based on our investigations, results and observations of many experiments, our presented models yielded outstanding performance and accuracy within our investigative framework of the NB-IoT technology.

http://doi.org/10.18080/jtde.v10n3.532

The contributions of our work are as follows:

- We analyzed the performance of various DL models in terms of DoSl attacks detection accuracy in NB-IoT wireless access networks.
- We provided a simulation model that can be used to generate and extract a novel dataset for DoSl attacks in an NB-IoT network architecture.
- We showed through extensive simulations within the scope of our study and using the constructed simulation model that:
    1. RNN DL models are capable of successfully and reliably classifying traffic data into either a DoSl attack or a normal record with very high accuracies and that the performance of an LSTM classifier outperforms the performance of GRU by several orders of magnitude. However, an interesting finding was that, on a preprocessed version of traffic data, accuracies of both LSTM and GRU classifiers improved to 99.1% and 98.6%, respectively, making the performance roughly the same, which may justify the use of GRU instead of LSTM due to its higher performance-cost ratio and higher performance efficiency in terms of less memory usage and higher training speed.
    2. RNN DL models outperform some traditional ML models (support vector machine, Gaussian naïve-Bayes, and logistic regression) in terms of the accuracy of DoSl attacks' detection in NB-IoT wireless access networks.

The rest of the paper is organized as follows. Section II discusses relevant foundation, background and associated literature. We present in Section III our methodology and implementation, including system architecture. Simulation results are presented and discussed in Section IV. Finally, we draw conclusions in Section V.

Relevant Background

This section provides a review and discusses relevant background of the main components and underlying principles of the framework that constitutes this work.

Narrowband Internet of Things

Countless applications of the NB-IoT technology make such a network an attractive target for attackers to invade. Physical and logical channels for NB-IoT are constructed based on the LTE technology, but with some corresponding NB-IoT-specific variations and improvements (Wang *et al.*, 2017). For instance, in the group of *Downlink* channels, modifications include the definition of the Narrowband primary synchronization signal (NPSS), the Narrowband

http://doi.org/10.18080/jtde.v10n3.532

secondary synchronization channel (NSSS), the Narrowband physical broadcast channel (NPBCH), the Narrowband reference signal (NRS), and Narrowband physical downlink shared channel (NPDSCH). In the group of *Uplink* channels, there are the Narrowband physical random access channel (NPRACH) and Narrowband physical uplink shared channel (NPUSCH). These two channels are used most frequently in communications. Unlike LTE physical channels, these channels are multiplexed.

The NB-IoT technology is a centralized system and its standard was firstly started within the LTE network architecture. The evolved node B (eNB) controls downlink as well as uplink scheduling to ensure resource coordination between UEs. The NB-IoT uplink communication is initiated when any user equipment (UE) device requests a transmission to an eNB using the random access procedure (NB-IoT RA) (Martiradonna, Piro & Boggia, 2019), as shown in Figure 1. Different uplink physical channels are used during transmission, including NPRACH and NPUSCH. The eNB receives a transmission request (Random Access Preamble) and sends a scheduling grant (Random Access Response) indicating time and frequency allocation to the device. Downlink communication starts after the device completes sending its identity and other important information about its transmission. Different downlink physical channels are used, including the NPBCH, the NPDCCH, the NPDSCH, the NPSS, and the Narrowband reference signal (NRS) (Miao *et al.*, 2017).

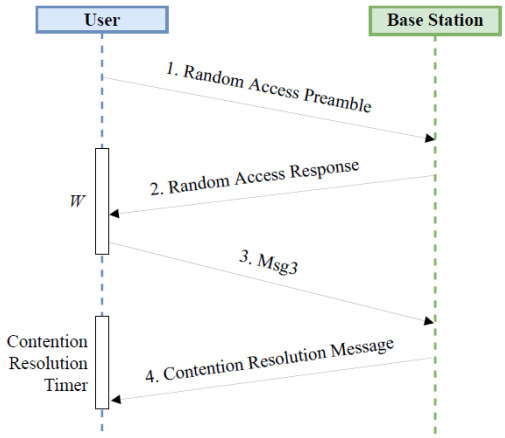


Figure 1. NB-IoT UE RA sequence diagram (Martiradonna, Piro & Boggia, 2019).

NB-IoT is a new fast-growing wireless cellular technology standard, first standardized in Release 13 of the 3GPP (Chen *et al.*, 2017; TR-45.820, 2015). NB-IoT has high energy efficiency and minimum power consumption that enables a battery life of more than 10 years. It supports a massive number of new connections, using a very small part of the available spectrum — as little as approximately 180 kHz — increasing capacity, while it is capable of an extended reach to underground and closed hard-to-reach spaces, providing deep indoor coverage (Chen *et al.*, 2017). NB-IoT can support a wide range of applications, which makes it such a promising technology that enables it to develop and expand several markets and organizations with

http://doi.org/10.18080/jtde.v10n3.532

increased efficiency (Xu & Darwazeh, 2018; Fattah, 2018). Nevertheless, in an NB-IoT network, all sensors' data are sent to the server through their nearest connected serving base station (eNB). The server, in turn, processes and retransmits the data. Normal traffic UE's data in a typical application is exchanged between server and the serving base station of the connected UE. However, attackers can communicate with other UEs and their coupled sensors in their base station's coverage area and attack them.

Denial-of-Sleep attacks

The main purpose of UEs and their associated sensors in current NB-IoT networks is the collection of data, where UEs or nodes are configured to go into sleep mode when their scheduled tasks are accomplished to increase lifetime of their batteries. As a consequence of several data collections by NB-IoT services, data disclosure risk exists during transfer on-network, and during data processing by various network elements. Data packets exchanged between various components throughout sessions, that are established between UEs and eNB base stations, could be captured and monitored using hijacking tools employed in the network data communication domain. In that case, the communication is seized, allowing attackers to evaluate security vulnerabilities within captured communication messages after extraction of data. The compromised NB-IoT UEs may induce a signalling storm due to shared mobile telecommunication networks with massive numbers of UEs and mobile telecommunications subscribers.

It is possible to forge or tamper with the core network signalling of a NB-IoT system; or it may be replay-attacked due to the lack of a mechanism for mutual authentication of network elements. Moreover, the interface between NB-IoT core network and the Internet could be impaired due to multiple attacks from the Internet.

The DoSl attack is a type of denial-of-service attack that can prevent UEs from going into sleep mode and saving their energy. This will decrease the battery's lifetime of the UE. These attacks will send more messages in order to keep the UEs awake and deplete their stored energy, until totally consumed, by being kept in this state, even when there is no actual traffic, causing the node to die (Kaur & Ataullah, 2014). In addition, due to the fact that, when bandwidth is fully occupied by DoSl attack traffic, a communications jamming effect will be inflicted upon normal data packets that are being exchanged, further indirectly inducing additional duty, such as that of listening or retransmissions, and hence more power dissipation (Brun *et al.*, 2018).

Different types of DoSl attacks could be applied considering the various classes of networks (Mahalakshmi & Subathra, 2014; Niu *et al.*, 2012; Yuan *et al.*, 2019). For instance, different

http://doi.org/10.18080/jtde.v10n3.532

types of *sleep-deprivation* attacks aim at maximizing the power consumption of UEs, which, in turn, reduces their lifespan. An attacker initiates an interaction with the target node, dragging out the interactions as long as possible, depriving the node from entering into sleep mode, dissipating much needed power that could be conserved. The *HELLO flood* attack relies on the fact that many routing protocols in a wireless sensor network require network nodes to announce themselves by broadcasting 'hello' packets. A *HELLO flood* attack occurs when a network is weighed down with packets trying to initiate connections; consequently, it can no longer respond to the requests.

This work investigates protection and flexible usage for NB-IoT networks, wherein the DoSl *HELLO flood* attack is implemented and deployed on the proposed design of a NB-IoT network. Machine learning is utilized for the detection of a DoSl attack.

Machine Learning

The process of choosing the most suitable ML algorithm involves several elements (Al-Rashdan & Tahat, 2020), which can influence our decision, since we will not be able to identify a single approach that will be most effective for all scenarios. ML algorithms are capable of discovering patterns and replicating them in a systematic way. ML algorithms are broadly categorized using learning methods, such as supervised learning, unsupervised learning, and reinforcement learning. A collection of supervised ML algorithms is employed and investigated. This is because the aim is to recognize a DoSl attack relying on previously collected training data. Their performance was evaluated and compared in detection of a DoSl attack in a simulated NB-IoT network topology and associated environment variables. We present below a brief introduction to the underlying approach and principles of operation for each of the employed ML methods and algorithms, including deep learning techniques, used for classification tasks in this paper. (Aggarwal *et al.*, 2018; Al-Rashdan & Tahat, 2020; Tahat *et al.*, 2021).

1) Support Vector Machine

Support Vector Machine (SVM) is used in regression and classification problems. In this algorithm, each data item will be plotted as a point in the n-dimensional space with the value of each feature being the value of a particular coordinate. Then, the classification will be performed by finding the hyperplane that differentiates the two classes very well (Burges, 1998).

2) Logistic Regression

Logistic regression (LR) (Le Cessie & Van Houwelingen, 1992) is used in solving regression and binary classification issues with one or more attributes to predict the target. It analyses the

http://doi.org/10.18080/jtde.v10n3.532

relationship between attribute variables and response variables (normal and abnormal traffic) using the Bernoulli distribution and probability.

3) Gaussian Naïve-Bayes

Gaussian Naïve-Bayes (John & Langley, 2013) is used for binary classification issues based on using Bayes Theorem while incorporating the assumption that each pair of features is independent.

4) Recurrent Neural Networks

A recurrent neural network (RNN) is a type of deep learning neural network, subdivided into long short-term memory (LSTM) and gated recurrent unit (GRU), which will be discussed in greater detail subsequently.

Deep Learning

Attaining high accuracy and large success rates in a wide range of applications in various fields requires using deep learning ML techniques and algorithms in conjunction with big data analysis. Deep learning (DL) is a subcategory of ML that performs the machine-learning tasks, relying on its own learnt experience without explicit programming, thereby extracting valuable patterns from the involved dataset. Hence, DL can break the limitations of other ML methods on extracting well-represented features. It allows for composing multiple sequences from that input. It uses and processes the previous input by sending feedback signals to compute the output.

RNNs are different from other neural network types, since RNNs have memory elements as part of their architecture. Hence, RNN DL neural networks have attracted significant attention for applications involving sequential tasks (Tang *et al.*, 2019). Working with an RNN implies that the dataset incorporates sequential information necessary to solve the presented issue or problem (Tang *et al.*, 2019).

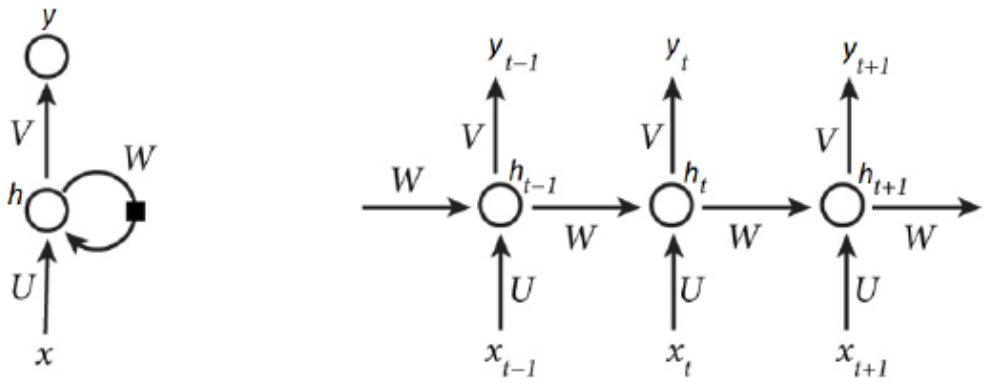


Figure 2. Architecture of a Recurrent Neural Network (Wei & Nguyen, 2019).

http://doi.org/10.18080/jtde.v10n3.532

As shown in Figure 2, a basic RNN consists of three basic layers with feedback that serves as memory, where the first layer is input, the second layer is the hidden layer, and the output layer is a third layer. In RNNs, neural sequence connections between units are formed as a directed cycle. The hidden states in RNNs are computed using a function that takes sequence inputs, $x = (x_1,.., x_T)x = (x_1,.., x_T)$, and uses them with internal memory to exhibit temporal behaviour. It also incorporates a sequence of hidden layers, $h = (h_1,.., h_T)h = (h_1,.., h_T)$, to generate an output vector, $y = (y_1, \dots, y_T)y = (y_1, \dots, y_T)$, at time T,T, which is the last step in time in the sequence (Wei & Nguyen, 2019). RNN is computed based on the previous hidden state $h_{t-1}h_{t-1}$ with input at the current step x_tx_t to calculate the current hidden step h_t, h_t,using (1):

$$h_t = g(Ux_t + Wh_{t-1})h_t = g(Ux_t + Wh_{t-1}) \quad (1)$$

RNN has processing layers to learn the representations of data with many levels of abstractions since it builds many hidden layers with multiple simple features to represent a developed concept. In addition, it has a several hidden layers that repeat the learning process many times, which makes DL much more powerful. In brief, DL is a subset of ML that uses artificial neural networks having three or more layers with nonlinear processing units to perform enhanced learning from large volumes of data (Al-Rashdan & Tahat, 2020; Chaabane *et al.*, 2020). To this end, in our investigation, DL proved to be very advantageous in the detection of DoSl attacks on NB-IoT networks.

Recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are DL neural networks. CNN is a multi-layered neural network with a unique design to extract complex specific data features in each layer by deriving the most important information from the input data (Muhammad *et al.*, 2018). On the other hand, the RNN-type DL neural network receives an input sequence and generates the output, where U and W are the weight matrices of the network, and $g(f)$ is the nonlinear activation function, such as a *sigmoid*. As a final stage to find y, the previous outputs are re-formed by using another activation function V, such as *softmax*, in order to implement and represent a linear output in the final layer. The RNN DL neural networks are further subdivided into two architectures: the long short-term memory (LSTM) and the gated recurrent unit (GRU), that are briefly introduced below.

1) Long Short-Term Memory

Linear long short-term memory (LSTM) is an RNN-specific architecture capable of learning long-term dependencies and remembering the information for prolonged periods of time as a default. This LSTM architecture is organized in a chain structure and has additional interactions per module (or cell), where each cell has three gates. These gates are input gate (i_ti_t), which determines the ratio of the input when calculating the cell state, the forget gate (f_tf_t), which decides on passing on or forgetting the previous memory h_{t-1}, and the output

http://doi.org/10.18080/jtde.v10n3.532

gate (o_t), which decides if it should pass the output of the memory's cell or not. In addition, there is a cell activation vector (c_t). All of these parameters can be calculated using (2)-(6), respectively (Wei & Nguyen, 2019; Kim *et al.*, 2016; Li *et al.*, 2018; Zadeh *et al.*, 2010).

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + V_i c_{t-1}) i_t = \sigma(U_i x_t + W_i h_{t-1} + V_i c_{t-1}) \quad (2)$$

$$f_t = \sigma\big(U_f x_t + W_f h_{t-1} + V_f c_{t-1}\big) f_t = \sigma\big(U_f x_t + W_f h_{t-1} + V_f c_{t-1}\big) \quad (3)$$

$$c_t = f_t c_{t-1} + i_t tanh(U_c x_t + W_c h_{t-1}) c_t = f_t c_{t-1} + i_t tanh(U_c x_t + W_c h_{t-1}) \quad (4)$$

$$o_t = \sigma(U_o x_t + W_o h_{t-1} + V_o c_t) o_t = \sigma(U_o x_t + W_o h_{t-1} + V_o c_t) \quad (5)$$

$$h_t = o_t tanh(c_t) h_t = o_t tanh(c_t) \quad (6)$$

where σ is the logistic *sigmoid* function (7), and U, W, and V are the weight matrices of the peephole connections. The second phase is to use forget gate (f_t), which takes the c_t and decides what parts to keep and to forget. This process of identifying and excluding data is decided by the logistic sigmoid function formulated by (7), which takes the output of the last LSTM unit (h_{t-1}) at time $t-1$ and the current input (x_t) at time t. Additionally, the sigmoid function determines which part from the old output should be eliminated.

$$\sigma(s) = \frac{1}{(1+e^{-s})} \quad (7)$$

The third phase takes the c_{t-1} coming from the forget gate, and the h_{t-1} coming from the first gate, and simply combines them. In the final step, the output value (h_t) is based on the output cell state (o_t) but in a filtered version.

2) Gated Recurrent Unit

The gated recurrent unit (GRU) is a newer generation of RNN. The special architecture of GRU uses fewer gates than LSTM; therefore, training time of GRU is shorter than training time in LSTM. In addition, GRU has fewer parameters than LSTM, as it lacks an output gate, and has only a reset gate and update gate. LSTM and GRU and their different properties will be used in this paper to detect DoSl attacks in the NB-IoT access technology.

Related work

Many studies in recent years (Tang *et al.*, 2019) have proposed the use of GRU, SVM, Naïve-Bayes and LSTM (Wei & Nguyen, 2019) in intrusion detection systems, where GRU has achieved the best accuracy when applying these methods on different datasets, such as the NSL-KDD dataset. In this work, these models will be applied and investigated on a simulated NB-IoT network using the presented NB-IoT characteristics, that are based on those of LTE built-in designs in OPNET and the LTE-Sim tool (Miao *et al.*, 2017; Martiradonna *et al.*, 2018).

http://doi.org/10.18080/jtde.v10n3.532

Consequently, we have generated a new dataset specific to NB-IoT access technology network to perform our investigation by applying the presented models to our dataset. In other recent works, NB-IoT systems have been simulated in Hassoubah, Solaiman & Abdullah (2015) using network simulator-3 (ns-3) based on the srsLTE file. They used the random access procedure in their configurations, where their technique aimed to improve average access delay (time interval). They explained in detail how random access works, provided explanations of LTE and NB-IoT channels with their setup parameters, and, in addition, how they extracted channel codes using ns-3 LTE-Helper.

Multiple models have been put forward to detect and mitigate DoSl attacks. Hassoubah, Solaiman & Abdullah (2015) discuss three models to detect attacks, such as absorbing Markov chain (AMC), secure wake-up scheme and isolation table intrusion detection system (ITIDS). A study in Saeedi (2019) presented methods for distributed denial-of-service (DDoS) attack detection in NB-IoT network, where ML algorithms were then applied to return one working memory instead of a pair of long-term and short-term memories. GRU cells have two input features, input vector Xt and previous output vector $h(t\text{-}1)$. GRU has gates that perform logical operations in addition to nonlinear transformations, in order to calculate the output of each gate. Equations (8)-(11) describe relationships between input and output (Wang, Liao & Chang, 2018).

$$r(t) = \sigma_g(W_r x(t) + U_r h(t-1) + b_r) \tag{8}$$

$$z(t) = \sigma_g(W_z x(t) + U_z h(t-1) + b_z) \tag{9}$$

$$h(t) = \big(1 - z(t)\big) \circ h(t-1) + z(t) \circ h(t) \tag{10}$$

$$h(t) = \sigma_h\big(W_h x(t) + U_h(r(t) \circ h(t-1) + b_h)\big) \tag{11}$$

where $z(t)$ is called the update gate vector, $r(t)$ is called the reset gate vector, W and U are defined to be parameter matrices and vectors. In addition, $h(t)$ is set to be an activation function, σ_g is the sigmoid function, and σ_h is the hyperbolic tangent. The element-wise product is used to distinguish between DDoS packets and normal packets. Hasan *et al.* (2019) have presented in their study the application of preprocessing and cleaning stages on a downloaded IoT dataset from the Kaggel website, so that it can be used for intrusion detection system design through applying various ML algorithms, including random forest, SVM and logistic regression. A genetic algorithm-based approach has been presented in Gunasekaran & Periakaruppan (2017) for DoSl attack detection in a wireless sensor net environment.

http://doi.org/10.18080/jtde.v10n3.532

Methodology and Implementation

A NB-IoT network was designed and simulated using the ns-3 network simulation tool. The cellular network parameters were configured to use the NB-IoT on a 5G network using this simulator. The set-up network was then attacked by a *HELLO flood* DoSl attack in order to derive a dataset for investigation, including training and testing, where the dataset would be applied to the implemented DL modules.

Network set-up

The attacker of the *HELLO Flood* attack in this experimental framework is assumed to be located within the area covered by the base station serving the NB-IoT network. It will launch an attack against other UEs (e.g., sensors) in the same network, when communicating through the serving base station. The attack traffic direction is depicted in Figure 3. Our DoSl attack detection module is placed in NB-IoT base station, so as to scan received traffic and classify whether it is normal/attack traffic to perform the detection.

As described previously, all UEs with different IP addresses send their data to the server through their nearest connected base station. However, the attacker broadcasts the flood of *HELLO* messages to the sensors served by the same network through its base station. Meanwhile, the designed NB-IoT network topology consists of one server in the network and three base stations, where each base station has up to 10 connected sensors or UEs. Three attackers (i.e., UEs that initiate an attack) are implemented in this network (one attack in each base-station service area). The NB-IoT network testbed architecture is depicted in Figure 4. This NB-IoT network was simulated based on values that are listed in Table 1.

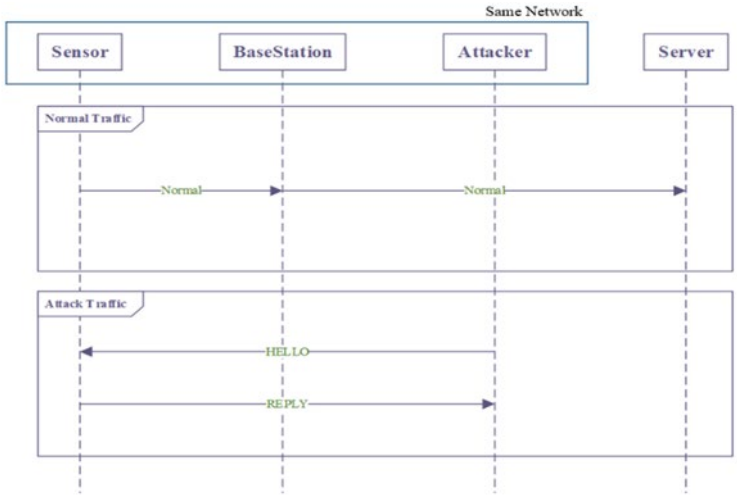


Figure 3. The attacker acts as a Man-in-the-Middle and exchanges HELLO message/reply with UEs. The server and UEs exchange normal messages through the base station.

http://doi.org/10.18080/jtde.v10n3.532

Table 1. NB-IoT network simulation parameters.

Parameter	Value
Number of base stations (eNB)	3
UE nodes (sensors)	10 nodes per cell
Distance between UE nodes	300 m
Distance between nodes and base station	500 m-1000 m
Tx Power	500 mW
Rx Power	80 mW
Network access procedure	Random Access

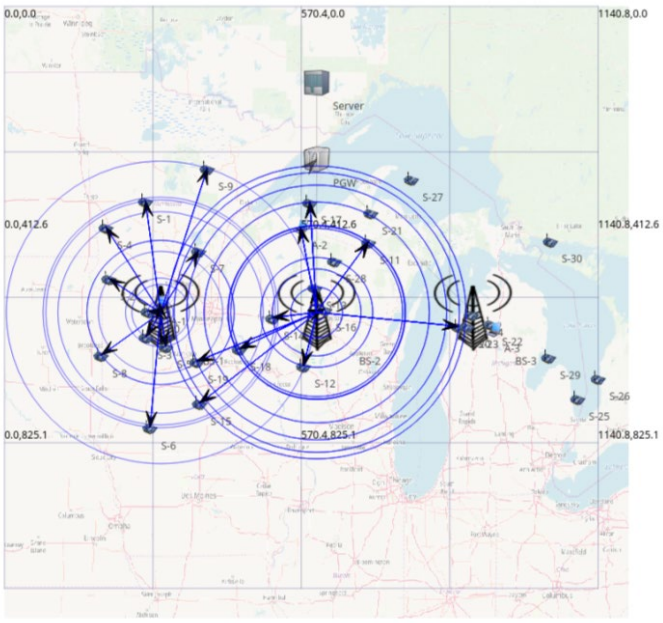


Figure 4. Designed NB-IoT network testbed for considered topology.

Dataset generation

After simulating the NB-IoT network topology, the *HELLO flood* attack was implemented and applied as a DoSl attack among the UEs. In order to improve battery autonomy, the NB-IoT standard includes Power Saving Mode (PSM) parameter configurations, which allow the device to enter into a deep sleep mode ranging from seconds to days. In this sleep mode, the network is no longer connected to the device, where the use of sleep mode is a matter of choice over power consumption and device reachability (Ehsan & Khan, 2012). However, attackers send a flood of packets in order to get the sensor out of this sleep mode. Sensors are configured to be in sleep mode. When they receive the *HELLO* message request, they will wake up and reply with *HELLO*-REPLY messages, consuming power each time they wake up and reply with *HELLO*-REPLY.

A *HELLO flood* DoSl attack can be induced by a node (the attacker), which broadcasts a *HELLO* packet with a very high power in order to attack a large number of nodes in the NB-IoT network in the same coverage area. These nodes are then convinced that the attacker node is their own trusted neighbour, so that the nodes will respond to the *HELLO* messages and waste their

http://doi.org/10.18080/jtde.v10n3.532

energy. Consequently, the network is left in a state of confusion (El Soussi *et al.*, 2018). Whenever the attacker starts the *HELLO Flood* DoSl attack, sensors are not able to enter into sleep mode because they have received a large number of HELLO messages. As a result of this communication, the energy of the UE will be consumed rapidly and eventually the sensors will die due to drained batteries.

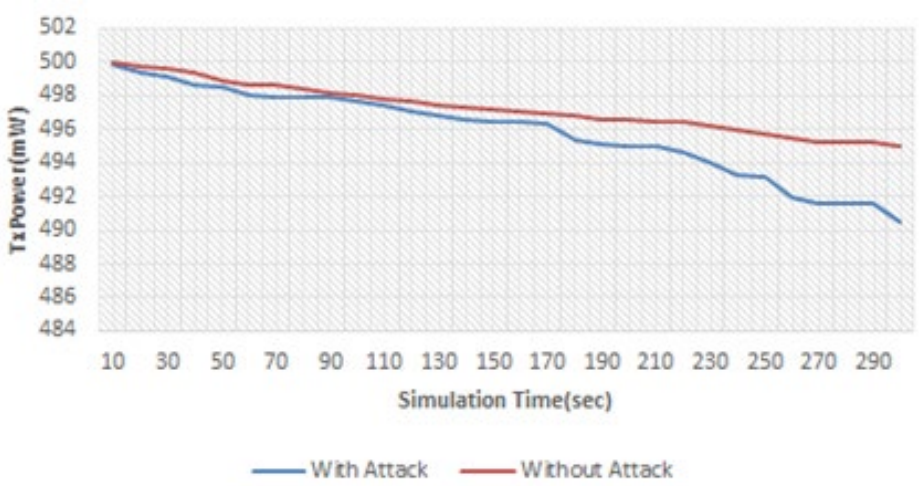


Figure 5. Power dissipation comparison on the node with/without applying DoSl attack on Tx Power.

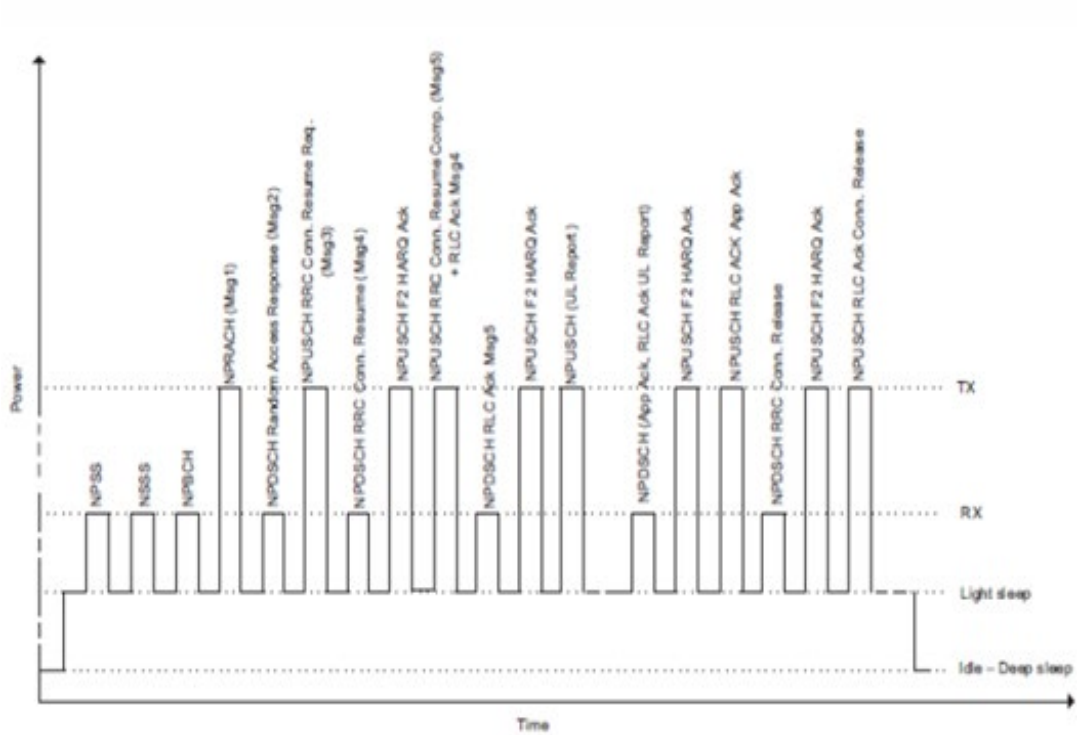


Figure 6. NB-IoT device power consumption in different modes (Liberg *et al.*, 2017).

A plot comparing power dissipation between a normal node and a DoSl attacked node during transmission (Tx) is depicted in Figure 5. The power dissipation reached up to 10 mW in 280 s, when transmitting packets of HELLO-REPLYs, where the power dissipation reached up to 3 mW in 280 s, when receiving packets of HELLO-REQUESTs for the same attack criteria. This dissipation of power demonstrates that Tx channels in NB-IoT consume more power than

http://doi.org/10.18080/jtde.v10n3.532

receive (Rx) channels, when the node tries to reply with a HELLO-REPLY message. Using transmission channels discussed before, including NBRACH and NPUSCH transmission channels, power consumption during different states of a NB-IoT device is shown in Figure 6 (Liberg *et al.*, 2017).

The ns-3 simulation was run for five continuous days of the designed NB-IoT network with the implementation of the *HELLO flood* DoSl attack on the network, in order to generate and export our dataset for investigation, analysis and performance evaluation.

Preprocessing of the dataset

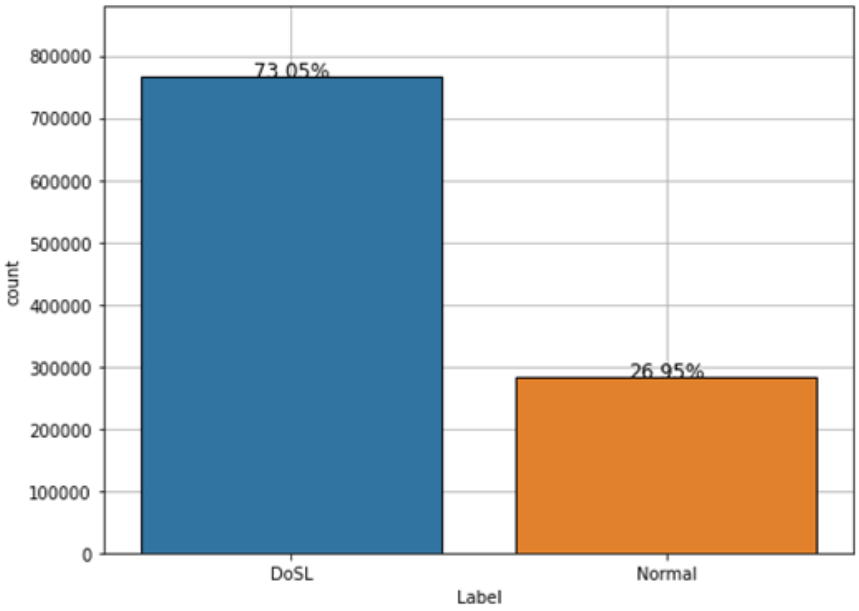


Figure 7. Bar chart representation for the distribution of target labels in dataset.

This DoSl attack detection dilemma can be treated as a binary classification problem. Therefore, when generating the dataset, it was designed to have two labels, namely: *Normal* or *Attack* labels. The distribution of *Normal* and *Attack* packets in the whole dataset is depicted in Figure 7.

The blue bar shows the packet rate of the *Attack* packets in the generated dataset, which is 73.05%. The orange bar shows the packet rate of *Normal* packets, which is 26.95% in the dataset that was generated from the DoSl-attacked NB-IoT network topology, as was discussed before. The dataset will be preprocessed in order to be more suited for presentation to each of the collection of investigated ML models. The next two steps will be performed in order to select which features in the generated dataset have more relevance in better detecting the *HELLO flood* DoSl attack.

http://doi.org/10.18080/jtde.v10n3.532

1) Multi-Correlated Features

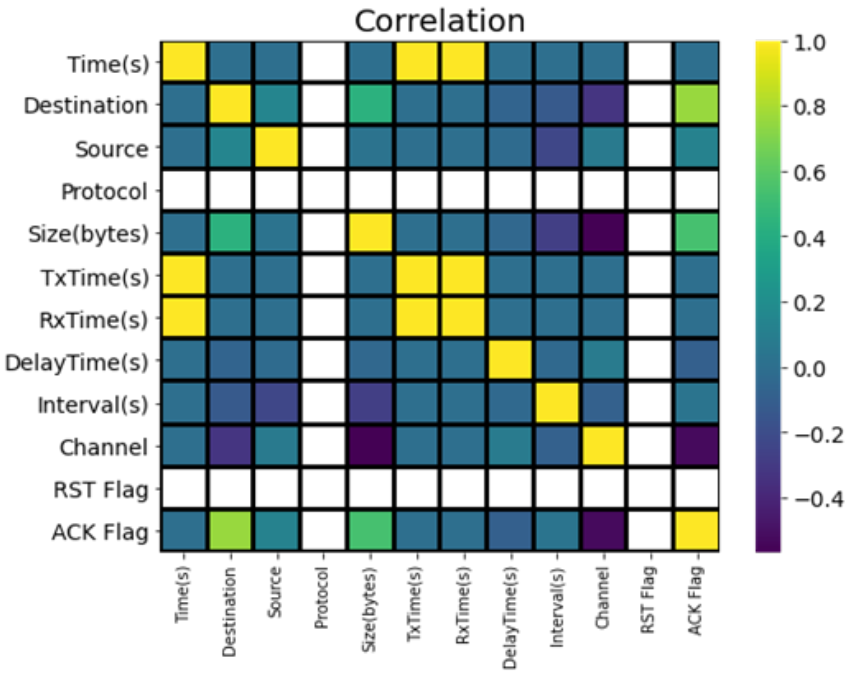


Figure 8. Heat map for Pearson correlation values between features in the extracted NB-IoT DoSl dataset.

Multi-Collinearity means that there are many features in the dataset that are highly correlated with each other, so they are monotonically increasing or monotonically decreasing together. In order to find the correlation, the Pearson correlation coefficient was used to extract the main features that are least correlated to each other. The linear correlation between two variables x and y could be measured by using (12), as in Wang, Liao & Chang (2018):

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \acute{x})(y_i - \acute{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \acute{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \acute{y})^2}} \quad (12)$$

where $\acute{x}$ is the mean value of x, $\acute{y}$ is the mean value of y, and r_{xy} is the Pearson coefficient between x and y. The results of applying (12), for the correlation matrix between each pair of features, are illustrated in Figure 8.

2) Constant Features

Constant features increase the redundancy in the dataset, which affects the detection accuracy. For this reason, they should be removed. In order to remove these features, the standard deviation has been calculated for each feature column in the dataset (Sharafaldin, Lashkari & Ghorbani, 2018).

3) Normalization

A scaling method will be applied in order to obtain the appropriate ranges of the feature values. For further applications of the dataset in DL models, it is required to perform normalization in order to make the samples more comparable. We mapped the original data range (X) into

http://doi.org/10.18080/jtde.v10n3.532

another data scaled range by taking the minimum value (X_{min}) and the maximum value (X_{max}) and replacing them with the chosen new scaled range ($X_{Normalized}$). Then, we computed the normalized values of the samples using (13) (Yin *et al.*, 2017).

$$X_{Normalized} = \frac{(X - X_{min})}{(X_{max} - X_{min})} \quad (13)$$

Our dataset has then been divided into training, validating and testing ranges of samples, as demonstrated in Table 2, in order to investigate, analyze and evaluate performance in detection accuracy of our collection of considered ML models through computation of evaluation metrics for these models.

Table 2. Number of records used in paper.

Kind of record	Number of records	Used percentage
Training Records	734,002	70%
Validation Records	157,286	15%
Testing Records	157,286	15%

Numerical Results and Discussion

The machine learning RNN models were implemented using Python on the cloud-based environment, Google Colaboratory (Colab) platform (Google Colaboratory (Colab), 2021; Bisong, 2019), for training and testing of all samples to obtain reliable, consistent, and accurate results for verification. During the implementation of LSTM and GRU classifiers, a number of parameters were tuned so as to get the final design of the DoSl attack detection system given in Table 3. There are test values of the hyper-parameters that were based on ones adapted from Tang *et al.* (2019), Wei & Nguyen (2019), and Wang, Liao & Chang (2018), to start-off with. Nevertheless, the final retained values selected for the classifiers' design parameters were those that yielded best achieved accuracy.

Table 3. Design Parameters of LSTM and GRU classifiers.

Parameter	Tested values	Best chosen values
Batch-Size	32, 64, 128	128
Hidden Layers	1, 2, 3	3
Optimizers	Adam, SGD.	Adam
Activation Functions	Tanh, Sigmoid, ReLU	ReLU, Sigmoid
Epochs	100	100

In order to evaluate the adopted ML models' effectiveness in attack detection, some evaluation metrics have been computed, such as loss function, confusion matrix parameters (Saeedi, 2019; Yin *et al.*, 2017), error matrix, receiver operating characteristic (ROC) curve, area under curve (ROC-AUC), precision, recall and $F_{1-score}$,to figure out the performance of the adopted LSTM and GRU models.

http://doi.org/10.18080/jtde.v10n3.532

Initially, the ML classifiers, including DL models, were tested on the original dataset (without preprocessing) to establish a performance baseline for comparison with the performance results achieved in application and testing with the preprocessed dataset.

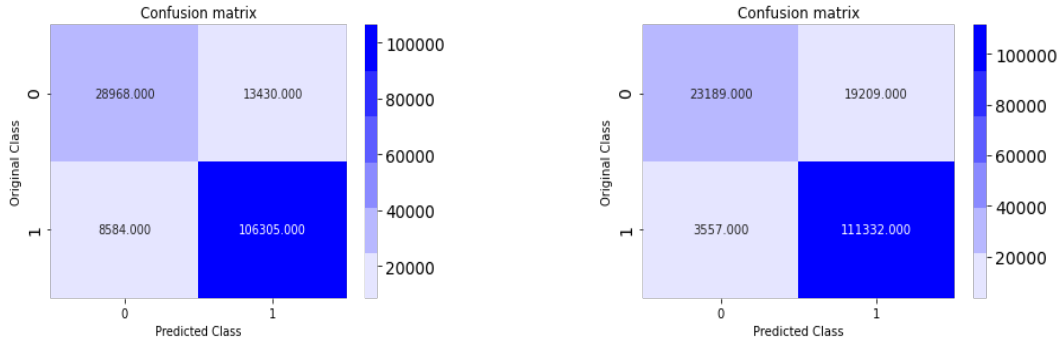
1) Original Non-Preprocessed Dataset Performance Evaluation Results

This original raw dataset has been fed into RNN models before preprocessing, in order to verify that the applied preprocessing steps have indeed enhanced their corresponding performance. As depicted in Table 4, the LSTM model has achieved 87%, 0.8042, 0.8878, 0.9252 and 0.9061, in terms of accuracy, ROC-AUC, precision, recall and $F_{1-score}$, respectively. The GRU has achieved 85.52%, 0.7579, 0.8528, 0.9690 and 0.9072, in terms of accuracy, ROC-AUC, precision, recall and $F_{1-score}$, respectively.

Table 4. Performance comparison in terms of accuracy, AUC, precision, recall, and F1-Score between LSTM, GRU, SVM, Logistic Regression and Gaussian Naïve-Bayes when applied on original dataset.

Model used on the original dataset	Accuracy	ROC-AUC	Precision	Recall	F1-Score
LSTM	0.87	0.80	0.89	0.93	0.91
GRU	0.86	0.76	0.85	0.97	0.91
Linear SVM	0.85	0.75	0.85	0.97	0.90
Gaussian Naïve-Bayes	0.78	0.71	0.85	0.85	0.85
Logistic Regression	0.85	0.76	0.86	0.95	0.90

The confusion matrices for both the LSTM and GRU models are illustrated in Figure 9, and the ROCs in Figure 10. In addition to the considered DL RNN models, performance metric values obtained for other traditional ML algorithms are presented in Table 4.



(a) (b)

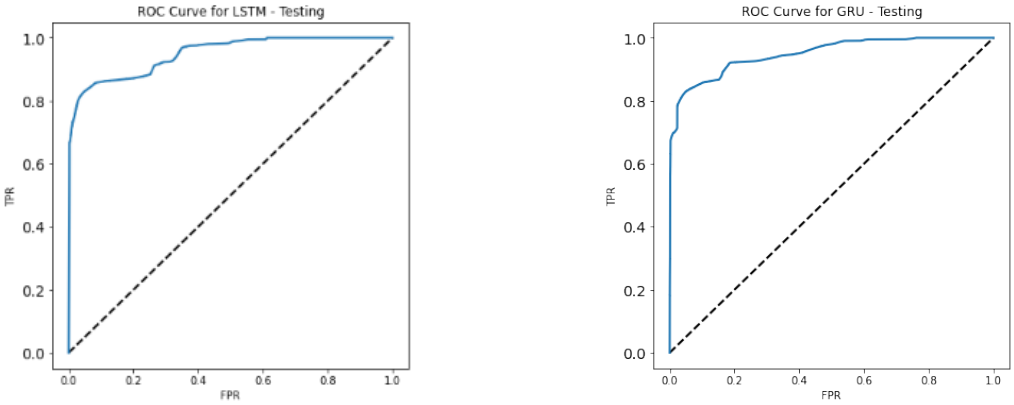
Figure 9. Confusion matrix for RNN models with original dataset: (a) LSTM; (b) GRU.

2) LSTM and GRU Performance Evaluation Results – The Preprocessed Dataset

The performance of LSTM and GRU models was compared after training the models for 100 epochs and evaluating the accuracy and confusion matrix parameters. As illustrated in Figure 11(a), the LSTM model has predicted successfully 113,881 true DoSl attack packets (labelled with class label 1), and predicted successfully 41,833 true normal packets (labelled with class 0 label),

http://doi.org/10.18080/jtde.v10n3.532

while it unsuccessfully predicted 1,008 attack packets and 565 normal packets, out of 157,286 total testing samples.



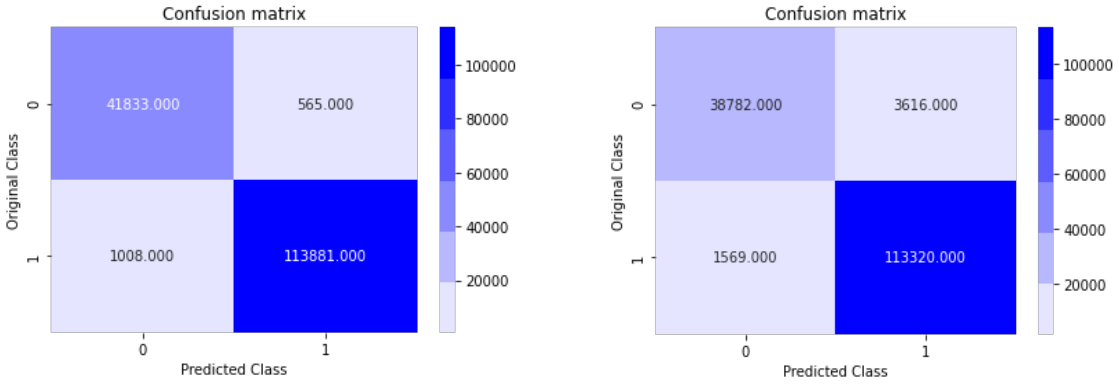
(a) (b)

Figure 10. ROC for (a) LSTM (b) GRU models when applied on original dataset.

The GRU model has successfully detected 113,320 DoSl attack packets and missed detection of 1,569 packets, while it has successfully predicted 38,782 normal records, as shown in the confusion matrix of Figure 11(b). While it has incorrectly predicted 3,616 packets out of 157,286 total testing samples. It is evident from the previous results that both RNN models have effectively learnt how to classify DoSl attack and normal packets in the preprocessed dataset. To support our observations, however, we evaluated other previously presented performance metrics. The loss function (LF) and accuracy plots versus epoch number for each of the LSTM and GRU models are illustrated in Figure 12 when training and validating the dataset when presented with the pre-processed dataset. The LF for both models is computed by the mean squared error (MSE) using (14):

$$LF = MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \quad (14)$$

where Y_i is target and $\hat{Y}_{i_i}$ is output.



(a) (b)

Figure 11. Confusion matrix for RNN models with preprocessed dataset: (a) LSTM (b) GRU.

Applying testing samples after concluding the training phase, the accuracy reached up 98.99% with a detection time of 2.54×10^{-5} second/record at the 100th iteration for the LSTM model,

http://doi.org/10.18080/jtde.v10n3.532

as shown in Figure 12(a). The GRU model has begun with a low value of accuracy, then it has improved along epochs until it has achieved an accuracy of 96.70%, with $1.90e^{-5}$ seconds/record detection time, as shown in Figure 12(b).

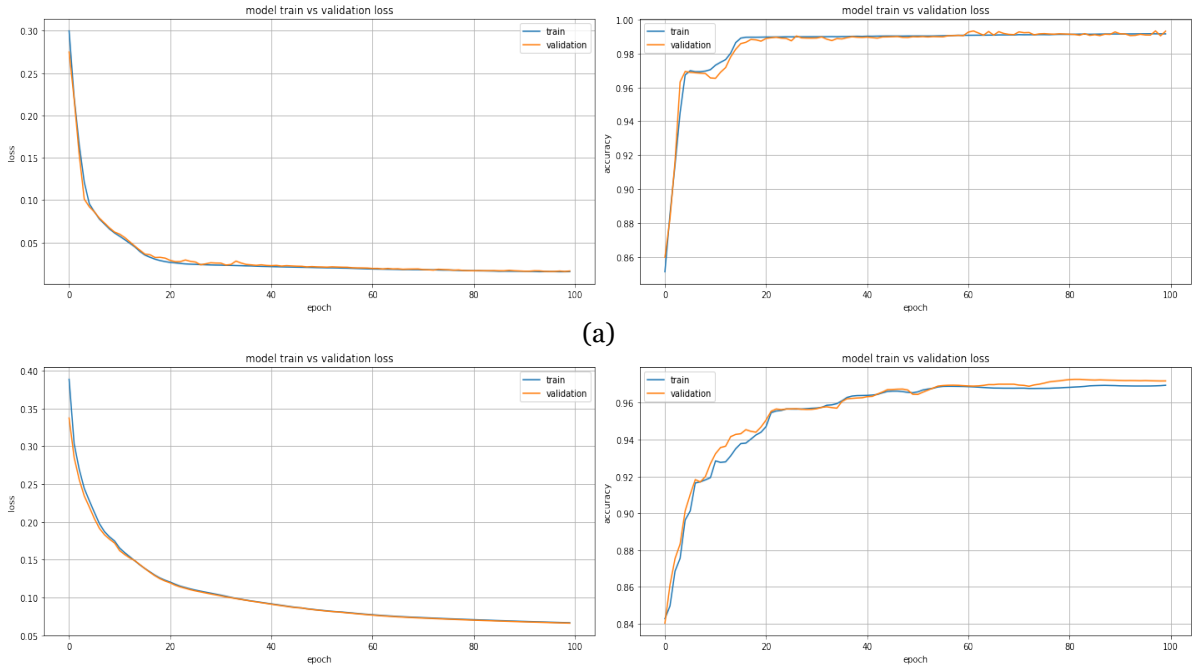


Figure 12. Plots of loss function and accuracy for training and validation range of dataset along epochs for (a) LSTM and (b) GRU models with preprocessed dataset.

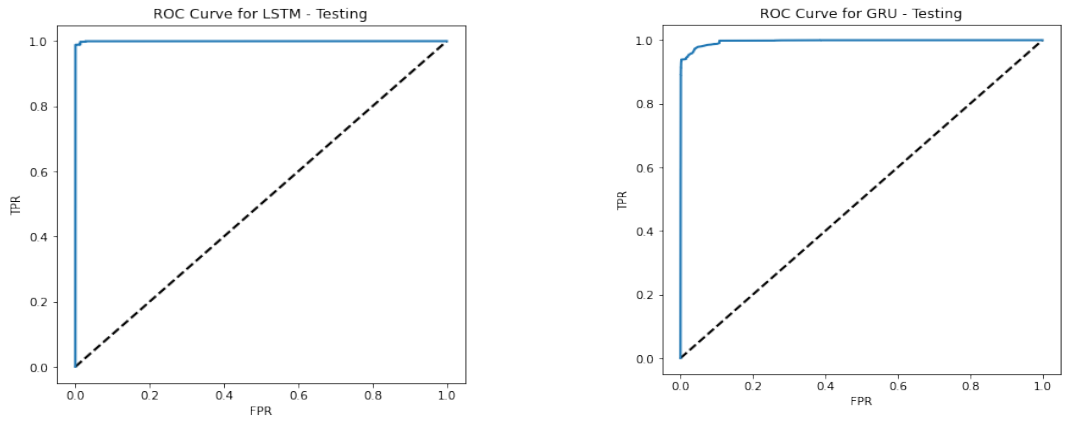


Figure 13. ROC-AUC for (a) LSTM and (b) GRU models when applied on preprocessed dataset.

The AUC is considered to be a powerful indicator of classifying performance of binary predictive models. As AUC reaches 1, the predictive performance of the model is considered optimal. The ROC-AUC for the LSTM and the GRU models are plotted in Figures 13(a) and 13(b), respectively, where it reached up to 0.9889 for LSTM and 0.9505 for the GRU. Also, the precision, recall, and $F_{1-score}$ performance metrics have been calculated for both models, where the LSTM model attained the respective values of 0.9950, 0.9912 and 0.9931, while the GRU model achieved 0.9690, 0.9863, and 0.9776 in terms of precision, recall, and $F_{1-score}$, respectively, as detailed in Table 5.

http://doi.org/10.18080/jtde.v10n3.532

Table 5. Performance Metrics Results Summary for RNN Models and Traditional ML Algorithms.

Model Type	Accuracy	ROC-AUC	Precision	Recall	F1-Score	Detection time (seconds/ record)
Original dataset/LSTM	87.00%	0.8042	0.8878,	0.9252	0.9061	4.45×10^{-5}
Original dataset/GRU	85.52%	0.7579	0.8528	0.9690	0.9072	3.81×10^{-5}
LSTM	98.99%	0.9889	0.9950	0.9912	0.9931	2.54×10^{-5}
GRU	96.70%	0.9505	0.9690	0.9863	0.9776	1.90×10^{-5}
Linear SVM	85.10%	0.7549	0.8538	0.9682	0.9047	3.18×10^{-5}
Gaussian Naïve-Bayes	78.46%	0.7123	0.8514	0.8546	0.8472	1.27×10^{-5}
Logistic Regression	85.35%	0.7601	0.8572	0.9528	0.8991	1.27×10^{-5}

Achieved performance of the RNN models has also been compared to other traditional ML classification algorithms, including support vector machine (SVM), Gaussian naïve-Bayes (GNB), and logistic regression (LR). The SVM classifier model achieved an accuracy up to 85% with a DoSl attack detection response time of 3.18×10^{-5} seconds/record, when training 734,002 and testing 157,286 samples. The GNB and LR classifier models detected the DoSl attack by 78% and 85% in terms of accuracy, respectively. Their DoSl attack detection time response was 1.27×10^{-5} seconds/record for each model. The remaining considered performance parameters, including confusion matrix, precision, recall, and $F_{1-score}$, are also presented in Table 5. It is evident that, while DL RNN models have superior performance in terms of accuracy of detecting DoSl attacks, when compared with other well-known traditional ML algorithms, including SVM, GNB, and LR, within the framework of our investigation and analysis, the LSTM was the better performer than the GRU of the two RNN models.

Conclusion

The NB-IoT wireless connectivity technology within the broad domain of IoT facilitates expedited deployment, due to the special feature of flexible wide coverage, using a small frequency bandwidth on existing cellular technologies, with a nationwide or even global introduction, and integration with sustained seamless coverage of high capacity. Along with that, security challenges become more critical and worth investigating. In this paper, *HELLO flood* DoSl attack is investigated within the framework of a model NB-IoT network, where a novel dataset of 1,048,576 records has been generated for this purpose, utilizing the ns-3 network simulation suite. Our investigation revealed that DL RNN models, including LSTM and GRU, are very advantageous in the detection of DoSl attacks on NB-IoT networks.

http://doi.org/10.18080/jtde.v10n3.532

Also, when compared with other well-known traditional ML algorithms, including SVM, GNB, and LR, within the framework of our investigation and analysis, it was clear that DL RNN models have the superior performance in terms of accuracy in detecting DoSl attacks. Moreover, the LSTM model was a better performer than the GRU model among the two RNN DL models, where it has achieved a high detection rate up to 99% against the considered DoSl attacks. These results verify the feasibility of the proposed DL RNN models in enhancing network security of the numerous vertical industries that employ NB-IoT networks.

Acknowledgement

This work was completed as part of the thesis requirements for the degree of MSc in Electrical Engineering during the studies of Mrs. Tahani Bani-Yaseen at Princess Sumaya University for Technology.

References

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning*. Springer. https://doi.org/10.1007/978-3-319-94463-0

Al-Rashdan, W. Y., & Tahat, A. (2020). A comparative performance evaluation of machine learning algorithms for fingerprinting based localization in DM-MIMO wireless systems relying on big data techniques. *IEEE Access*, *8*, 109522–109534.

Bisong, E. (2019). Google Colaboratory, pp. 59–64 in: Bisong, E., *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress. https://doi.org/10.1007/978-1-4842-4470-8

Brun, O., Yin, Y., Augusto-Gonzalez, J., Ramos, M., & Gelenbe, E. (2018). IoT attack detection with deep learning. In ISCIS Security Workshop. Available at https://hal.laas.fr/hal-02062091

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, *2*(2), 121–167.

Chaabane, M., Williams, R. M., Stephens, A. T., & Park, J. W. (2020). circDeep: deep learning approach for circular RNA classification from other long non-coding RNA. *Bioinformatics*, *36*(1), 73–80.

Chen, M., Miao, Y., Hao, Y., & Hwang, K. (2017). Narrow Band Internet of Things. *IEEE Access*, *5*, 20557–20577. https://doi.org/10.1109/ACCESS.2017.2751586

Ehsan, H., & Khan, F. A. (2012). Malicious AODV: implementation and analysis of routing attacks in MANETs. In 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, 1181–1187.

El Soussi, M., Zand, P., Pasveer, F., & Dolmans, G. (2018). Evaluating the performance of eMTC and NB-IoT for smart city applications. In 2018 IEEE International Conference on Communications (ICC), 1–7.

Fattah, H. (2018). *5G LTE Narrowband Internet of Things (NB-IoT)*. CRC Press.

Google Colaboratory (Colab). (2021). https://colab.research.google.com/notebooks/intro.ipynb (Accessed 12 August 2021).

Gunasekaran, M., & Periakaruppan, S. (2017). GA-DoSLD: genetic algorithm based denial-of-sleep attack detection in WSN. *Security and Communication Networks*, *2017*. https://doi.org/10.1155/2017/9863032

Hasan, M., Islam, M. M., Zarif, M. I. I., & Hashem, M. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things*, *7*, 100059. https://doi.org/10.1016/j.iot.2019.100059

Hassoubah, R. S., Solaiman, S. M., & Abdullah, M. A. (2015). Intrusion detection of hello flood attack in WSNs using location verification scheme. *International Journal of Computer and Communication Engineering*, *4*(3), 156. https://doi.org/10.17706/IJCCE.2015.4.3.156-165

John, G. H., & Langley, P. (2013). Estimating continuous distributions in Bayesian classifiers. arXiv preprint arXiv:1302.4964.

Kaur, S., & Ataullah, M. (2014). Securing the wireless sensor network from denial of sleep attack by isolating the nodes. *International Journal of Computer Applications*, *103*(1). https://doi.org/10.5120/18040-8920

Kim, J., Kim, J., Thu, H. L. T., & Kim, H. (2016). Long short term memory recurrent neural network classifier for intrusion detection. 2016 International Conference on Platform Technology and Service (PlatCon), 1–5. http://doi.org/10.1109/PlatCon.2016.7456805

Le Cessie, S., & Van Houwelingen, J. C. (1992). Ridge estimators in logistic regression. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *41*(1), 191–201.

Li, Z., He, D., Tian, F., Chen, W., Qin, T., Wang, L., & Liu, T. (2018). Towards binary-valued gates for robust LSTM training. In International Conference on Machine Learning, 2995–3004.

Liberg, O., Sundberg, M., Wang, E., Bergman, J., & Sachs, J. (2017). *Cellular Internet of things: technologies, standards, and performance*. Academic Press.

Mahalakshmi, G., & Subathra, P. (2014). A survey on prevention approaches for denial of sleep attacks in wireless networks. *Journal of Emerging Technologies in Web Intelligence*, *6*(1), 106–110. https://doi.org/10.4304/jetwi.6.1.106-110

Martiradonna, S., Grassi, A., Piro, G., Grieco, L. A., & Boggia, G. (2018). An open source platform for exploring NB-IoT system performance. In European Wireless 2018; 24th European Wireless Conference, 1–6.

Martiradonna, S., Piro, G., & Boggia, G. (2019). On the evaluation of the NB-IoT random access procedure in monitoring infrastructures. *Sensors*, *19*(14), 3237.

Miao, Y., Li, W., Tian, D., Hossain, M. S., & Alhamid, M. F. (2017). Narrowband Internet of Things: Simulation and modeling. *IEEE Internet of Things Journal*, *5*(4), 2304–2314. https://doi.org/10.1109/JIOT.2017.2739181

Muhammad, K., Ahmad, J., Mehmood, I., Rho, S., & Baik, S. W. (2018). Convolutional neural networks based fire detection in surveillance videos. *IEEE Access*, *6*, 18174–18183. https://doi.org/10.1109/ACCESS.2018.2812835

Niu, Y., Gao, D., Gao, S., & Chen, P. (2012). A robust localization in wireless sensor networks against wormhole attack. *Journal of Networks*, *7*(1), 187.

Popli, S., Jha, R. K., & Jain, S. (2018). A survey on energy efficient Narrowband Internet of Things (NBIoT): architecture, application and challenges. *IEEE Access*, *7*, 16739–16776. https://doi.org/10.1109/ACCESS.2018.2881533

Saeedi, K. (2019). Machine learning for DDOS detection in packet core network for IoT. Masters Thesis, Luleå University of Technology. Available at https://www.diva-portal.org/smash/get/diva2:1360486/FULLTEXT02.pdf

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), 1, 108–116. https://doi.org/10.5220/0006639801080116

Tahat, A., Awad, R., Baydoun, N., Al-Nabih, S., & Edwan, T. A. (2021). An Empirical Evaluation of Machine Learning Algorithms for Indoor Localization using Dual-Band WiFi. In 2nd European Symposium on Software Engineering, 1–6.

Tahat, A., Ersan, B., Muhesen, L., Shakhshir, Z., & Edwan, T. A. (2020). A compact 38 GHz millimetre-wave MIMO antenna array for 5G mobile systems. *Journal of Telecommunications and the Digital Economy*, *8*(3), 44–59. https://doi.org/10.18080/jtde.v8n3.299

Tang, T. A., McLernon, D., Mhamdi, L., Zaidi, S. A. R., & Ghogho, M. (2019). Intrusion detection in SDN-based networks: Deep recurrent neural network approach. In Deep Learning Applications for Cyber Security, 175–195. Springer.

TR-45.820. (2015). Cellular system support for ultra-low complexity and low throughput Internet of Things. V2.1.0. 3GPP. Available at https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2719

Wang, Y., Liao, W., & Chang, Y. (2018). Gated recurrent unit network-based short-term photovoltaic forecasting. *Energies*, *11*(8), 2163.

Wang, Y.-P. E., Lin, X., Adhikary, A., Grovlen, A., Sui, Y., Blankenship, Y., & Razaghi, H. S. (2017). A primer on 3GPP Narrowband Internet of Things. *IEEE Communications Magazine*, *55*(3), 117–123. https://doi.org/10.1109/MCOM.2017.1600510CM

Wei, F., & Nguyen, U. T. (2019). Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In 2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA), 101–109.

Xu, T., & Darwazeh, I. (2018). Non-orthogonal Narrowband Internet of Things: A design for saving bandwidth and doubling the number of connected devices. *IEEE Internet of Things Journal*, *5*(3), 2120–2129. https://doi.org/10.1109/JIOT.2018.2825098

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, *5*, 21954–21961. https://doi.org/10.1109/ACCESS.2017.2762418

Yuan, X., He, P., Zhu, Q., & Li, X. (2019). Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, *30*(9), 2805–2824. https://doi.org/10.1109/TNNLS.2018.2886017

Zadeh, M. R., Amin, S., Khalili, D., & Singh, V. P. (2010). Daily outflow prediction by multi layer perceptron with logistic sigmoid and tangent sigmoid activation functions. *Water resources management*, *24*(11), 2673–2688.