Journal of Autonomous Intelligence (2024) Volume 7 Issue 5
doi: 10.32629/jai.v7i5.1635

ORIGINAL RESEARCH ARTICLE

An efficient network optimized machine learning architecture framework for detection of malwares in IOT (NB-IOT) systems

R. Rajalingam*, K. Kavitha

Annamalai University, Chidambaram 608001, India

*** Corresponding author:** R. Rajalingam, sairamsai936@gmail.com

ABSTRACT

The Internet of things (IoT) is a wirelessly interconnected network of electrical gadgets. Due to their necessity of vast volumes of data in a single entity, the centralized machine learning (ML)-assisted systems that are widely used are difficult to use. Researchers and academics have a challenging issue about security and privacy in the IoT system. To overcome this issue, the paper presents Network optimised with machine learning classification is proposed. In terms of delay, security, accessibility, data transfer rate, energy consumption, spectral effectiveness, and coverage area, IoT and other wireless and mobile communication technologies function effectively. K-nearest neighbour (KNN) is one of the machine learning algorithms based on supervised learning technique is proposed. In specific applications, the proposed K nearest neighbour algorithm, is more accurate than existing methods such as decision tree (DT), random forest (RF), and support vector machine (SVM), and can be used to improve malware detection accuracy and also used to detect malware in NB-IoT. Using Aposemat IoT-23 datasets and assessment criteria including malware detection accuracy, recall, precision, and F1-score, the suggested technique was assessed. It was shown to be more accurate than competing methods and to increase security levels.

Keywords: Internet of things (IoT); machine learning (ML); network optimised; classification; K nearest neighbor (KNN); decision tree (DT); random forest (RF); support vector machine (SVM)

ARTICLE INFO

Received: 7 March 2024
Accepted: 26 March 2024
Available online: 21 August 2024

1. Introduction

New wireless network technologies are required for the Internet of things (IoT) to expand rapidly. Performance, heterogeneity, and large data processing have all significantly improved thanks to IoT[1]. High-data-rate services and low-data-rate services are the two broad categories into which IoT communication services may be divided from the standpoint of transmission rate[2]. This is also because IoT devices must meet certain specifications and have certain qualities, such low power consumption, extended range, affordability, and security. The efficient utilization of network resources for the IoT has sparked significant research and development efforts aimed at creating innovative systems and methods across all network layers[3].

Recently, some IoT communication technology has become matured and widespread. WAN and short-distance IoT connectivity methods are categorized by transmission distance[4]. Wi-Fi, Bluetooth, Zigbee, and other wireless technologies serve as examples of the former. Smart homes are where they are most often used. In low-data-rate services such as the smart parking mentioned before, which is typically referred to as LPWAN technology, the latter are sought[5].

The narrow-band Internet of things (NB-IoT), a critical low power wide area (LPWA) technology for intelligent low-data-rate applications, was developed by the third generation partnership project (3GPP)[6]. Intelligent environment monitoring and smart metering are popular uses. The components of the power grid are explained in depth in **Figure 1**. LPWAN is one of the topics that has attracted attention lately[7]. Internet of things IoT deployments are supported by a class of wireless communication technologies known as LPWA[8]. Wide area coverage, high connection density, ultra-low power consumption[9], and bidirectional triggering between the signaling plane and data plane are all features of the NB-IoT[10].

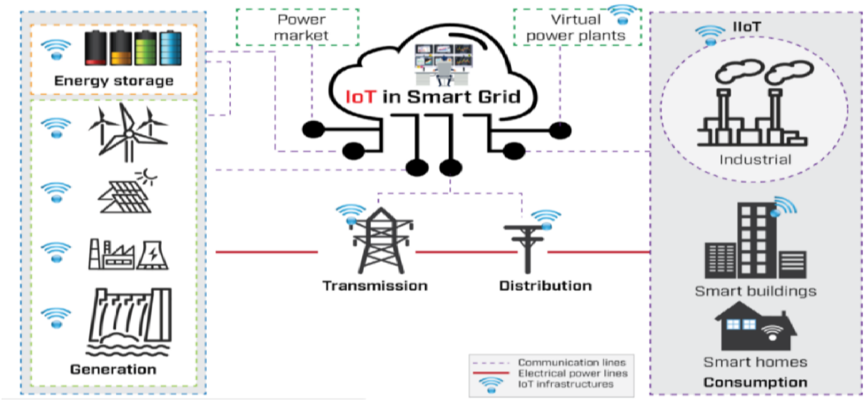

Figure 1. IoT based monitoring for power grid components.

In the existing work, IoT is mainly focused for malware detection, NB-IoT devices in heterogeneous contexts with varied network coverage make up the scenario. The current approach presents significant problems in terms of low power consumption, energy economy, data throughput, network dependability, and performance scalability. Along with the use of antidetection measures like packing and obfuscation, the quantity of new malware varieties is also boosting power consumption.

A novel approach for detecting malware in NB-IoT systems utilizing network-optimized classification was developed in the study to address this issue. In order for the NB-IoT requirements to be issued for standardization and commercialization, research efforts aimed at improving NB-IoT resource management must continue over the next half-decade. The simulation results, the optimized classification, in comparison with the current effort, to enhancing the NB-IoT malware detection and boost the scalability and reliability.

2. Literature review

Riaz et al.[11] presented an IoT malware detection using deep learning-based ensemble categorization techniques. The three stages of an approach are as follows: 1) The data is pre-processed using scaling, normalization, and denoising. 2) For the ensemble classifier to identify malware, one hot encoding is used, then the outputs from convolutional-neural-network (CNN) and long–short-term-memory neural network (LSTM) are picked. The suggested approaches perform better than the current methods on common datasets with an average accuracy when the simulation results are compared to state-of-the-art methods.

Jeon et al.[12] defined a dynamic analysis for Internet of things malware detection (DAIMD). By identifying both well-known and uncommon as well as variant IoT malware that has cleverly developed, the suggested technique lessens damage to IoT devices. The CNN model is used by the DAIMD technique to absorb IoT malware, which is then dynamically analyzed in a layered cloud environment. IoT malware is dynamically examined for properties related to the network, memory, process, virtual file system, and system call in a layered cloud environment. The CNN categorizes and trains IoT malware behavior photos after transforming behavior data into images. With the ability to see and comprehend the vast amount of behavioral data obtained via dynamic analysis, the performance outcomes may reduce the infection harm of IoT devices.

Asam et al.[13] proposed a convolution neural network (CNN)—based IoT malware detection architecture (iMDA). Multiple feature learning techniques are included in the proposed method's modular design, which is composed of the following building blocks: Three methods are employed to absorb a variety of information: CNN channel squeezing and boosting, edge exploration and smoothing, and multipath dilated convolutional operations. Edge and smoothing procedures carried out in the split-transform-merge (STM) block absorb local structural changes within malware classes. Recognizing the overall structure of malware patterns is done via multi-path dilated convolutional procedures. Additionally, channel merging and squeezing assisted in obtaining feature maps with a variety of variations and in controlling complexity. The input dataset is obtained from a Dell Core I i5-7500 with a GPU-enabled Nvidia GTX 1060 card and compared to multiple CNN designs. The suggested malware detection approach achieves measures like accuracy, F1-score, precision, Matthews Correlation Co-efficient (MCC), AUC-ROC and area under the precision and recall curve (AUC-PR). Strong discriminating abilities might be advantageous for future Android-based virus detection and IoT composite systems.

Dartel[14] proposed a new method for detecting malware by machine learning and the results are promising. Since IoT nodes often have subpar CPUs, the detection in a single IoT node has not yet been completed. By attempting to expand machine algorithms to the point where a single IoT device can carry out near real-time network traffic anomaly detection and label packets as "malware" or "benign," the potential for malware detection in a single IoT device is studied. The suggested machine learning technique is constructed using the IoT-23 dataset on an ESP32 device, which can categorize data points. IoT devices will be able to check for network connections during a malware attack or a "normal" as the simulation results when completely deployed.

Mustafa Hilal et al.[15] suggested a plan to identify malware in the IoMT while data is being transmitted that combines machine learning and blockchain technology. The three-step machine learning-based blockchain technology-malware detection scheme (MLBCT-Mdetect) is used: extraction, blockchain, classification. first step, by determining each feature's weight and minimizing the characteristics with lower weights, features may be extracted. The malware and benign nodes are separated out in the second stage using a support vector machine classifier. Additionally, the third stage enhances malware detection and dramatically increases speed and accuracy by using blockchain to record data of the chosen characteristics. With a low false positive rate and a greater genuine positive rate, the findings are very accurate.

Pei et al.[16] has proposed a new approach called FedMaIDE: a knowledge transfer technologies-based federated Internet of Things malware detection solution. With the use of a knowledge transfer mechanism, to infer labels toward unlabeled data, FedMaIDE accurately determines the underlying relationship between labeled and unlabeled records. In order to effectively capture various harmful behaviors, a specifically created subgraph aggregated capsule network (SACN) is employed. FedMaIDE's success in identifying IoT malware is shown by experimental findings on actual data, which also show how much privacy and robustness it can ensure.

Tamás et al.[17] suggested a SIMBIoTA, a novel malware detection method for IoT devices, was suggested. IoT devices with limited resources might benefit from similarity-based malware detection as it is fast and lightweight but also needs limited storage, and detects new, never-before-seen malware with surprising accuracy. The "National Research Council supported the Security Enhancing Technologies for the Internet of Things (SETIT) Project (2018-1.2.1-NKP-2018-00004)". Competitive detection and low resource needs were achieved.

HaddadPajouh et al.[18] suggested a recurrent neural network (RNN) deep learning has been suggested as a promising method for identifying IoT malware. Advanced RISC Machine (ARM)-based Internet of things applications are analyzed using RNN. Operation codes of the algorithms were trained on an IoT application dataset of 281 malicious and 270 benign software. The trained model is assessed with three alternative LSTM settings and 100 fresh IoT malware samples. The second configuration, which makes use of 2-layer neurons, is shown by the 10-fold cross-validation study to be more effective at identifying new malware types. The LSTM technique yields the best results, as shown by a comparison with different machine learning classifiers.

Mihoub et al.[19] developed two-part architecture for Denial-Of-Service (DoS)/Distributed Denial-Of-Service (DDoS) detection and mitigation. Fine-grained detection determines the attack type and packet type. This allows particular packet types to be mitigated. It assesses the multi-class classifier's "looking-back" detection component on the Bot-IoT dataset. A looking-back-enabled random forest classifier's accuracy is assessed.

Kaur et al.[20] suggested combining k-mean with firefly algorithms for anomaly identification. This approach employs classification to assess the test set and clustering to generate the training model. The NSL-KDD dataset yields outstanding results for the subject algorithm. K-means + bat, k-means, k-means++, canopy, and farthest initially were compared to the newly designed method. The findings suggest that k-means + firefly and k-means + bat outperform.

3. Proposed methodology

In this section, the distinguishing properties of NB-IoT, in order to shape later the problem statement. Further, it describes a basic security protocol that is: 1) Network optimizer malware detection in NB-IoT. 2) Detection of malware using machine learning classification methods.

3.1. Network optimizer

In general, the technology used to enhance network performance in any setting is referred to as "network optimization"[21]. This is significant in IT because the network is constantly being supplied with every device and apps generate vast data. The advantages of network optimization include increased application and network response times, quicker data rates, data recovery, the removal of duplicated data, and others.

3.1.1. Network optimization and IoT

Billion IoT devices will connect to the global network in the future, making network optimization crucial. Researchers and operators must develop ways to enhance IoT networks to lessen the impact of IoT traffic on other network services and effectively use network resources[22]. Due to the diversity of IoT applications and device kinds, the traffic produced by these devices differs from that of the cellular network. In order to monitor how IoT devices and services are functioning, IoT traffic also has to be controlled. To handle and enhance the communications from IoT devices' control plane, an effective technique is needed to alleviate this strain.

3.1.2. State-of-the-art solutions for IoT network optimization

To ensure that IoT networks operate at their best, several network optimization strategies have been developed. The network optimization strategy for the Internet of things is broken down into categories for your convenience [23]. An unprecedented quantity of data is being produced as a result of this new technology. The **Figure 2** depicts a categorization of network optimization objectives in the Internet of things.

Major concerns include data storage, routing, packet retransmission, data security, node mobility, and interoperability across diverse nodes. IoT devices need to emphasize energy efficiency to ensure reliable connectivity due to these demands. At the moment, 5% of all energy produced is used by the Internet. These issues must be addressed. With the goal of network optimization, this section discusses the objectives, strengths, and weaknesses of various works on network routing, energy conservation, congestion, heterogeneity, scalability, reliability, QoS, and security.

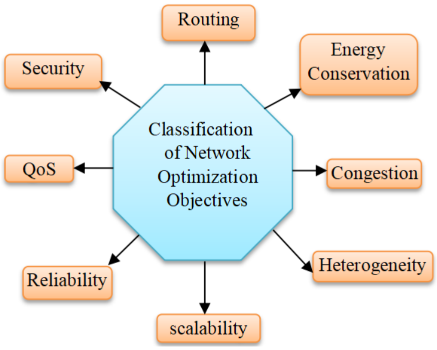

Figure 2. Classification of IoT network optimization objectives.

3.1.3. Security issues and challenges for network optimization in IoT

Security is essential for protecting networked data from breaches, thus it's best to offer an effective system. As it uses less energy, memory overhead, and latency is perfect for memory-constrained, energy-efficient Internet of Things devices. System power usage and heat production may be reduced thanks to this functionality. The security system offers a more effective method for managing keys, providing secure network access and preserving communication. This method is ideal for safeguarding IoT networks since it has a smaller memory footprint[24].

Any device linked to the Internet must have security in place owing to the increased risk of attack. People only have faith in a technology or product when it is sufficiently secure to resist malevolent activity for both the product and its data. Attackers may modify or damage IoT devices with poor security. For instance, a weakly protected smoke detection sensor linked to the Internet might be attacked with malware and transmit false/spam notifications or emails about its condition to its destination.

Issues:

- To protect IoT network data from many sorts of threats.
- Network exposure as a result of issues with the technology and how it was used.
- Assault using a side channel to enter the network.

Challenges:

- Network content, illegal resource access, and intruder prevention are part of IoT network security. Data should be safeguarded against hostile actions such man-in-the-middle attacks, denial-of-service (DoS) attacks, virus insertion, data eavesdropping, unauthorized system access, and so on since the network is designed to transmit data[25]. As a result, the problem is to present a single technique that protects against these types of attacks.
- Security vulnerabilities in networks occur for two reasons: security risks associated with the complete IoT network architecture and weaknesses in technology and protocol implementation and design[26]. Wireless networks become susceptible to malicious behavior or security breaches whenever devices are added to or removed from them, or both. With the use of this functionality, an intrusive party might distribute infected nodes amongst good nodes, hence decreasing network quality. As a result, the problem is to develop a security solution in this case.
- Attempts to get into the system by identifying weaknesses in the physical implementation of the encryption system, the system may be breached by the attacker thanks to factors including electromagnetic leakage, time information, energy utilization, and many more. As a result, the designer's challenge is to create a more powerful cryptography method.

3.2. Security detection of malware in NB-IoT using machine learning methods

To address these security issues, machine learning techniques are being used to improve the security services for NB-IoT. Following a discussion of the machine learning area, **Figure 3** explains the methods pertinent to this investigation [27]. In terms of accuracy, recall, and precision, these techniques perform better than decision tree, support vector machine, random forest, and k-nearest neighbors. Machine learning has emerged as a separate area of study within computer science as a result of the rapid growth of data mining methods and procedures. Every machine learning activity starts with the notion of training a model to do a task, such as classification, cauterization, regression, etc. The model is trained on the input dataset and predicts.

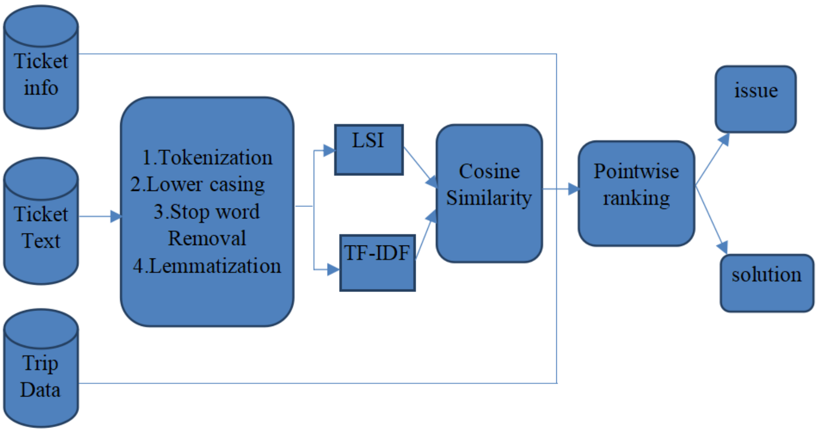

Figure 3. General workflow of the machine learning.

3.2.1. Supervised and unsupervised learning

Supervised learning and unsupervised learning are the two types of machine learning approaches. In supervised learning, tagged data serve as the basis for learning. The starting dataset in this instance maps data samples to the desired result[28]. There is no initial labelling of data in unsupervised learning, in contrast to supervised learning. Here, rather than attempting to forecast a number, the objective is to identify certain patterns in the collection of unsorted data.

3.2.2. Classification methods

Machine learning can classify or cauterize malware detection. Unknown malware kinds should be cauterized into multiple clusters depending on specific criteria that are found by the algorithm. Training a model on a large sample of hazardous and benign data simplifies categorization [29]. For known malware families, classification is more straightforward and precise than cauterization procedures.

3.2.3. K-nearest neighbors

Machine learning techniques that are both simple and accurate include KNN. Due to KNN's non-parametric nature, data structure is not assumed. Classification and regression problems may be tackled using KNN [30]. The k training examples closest to the input instance determine the prediction in both cases. **Figure 4** shows the KNN classification problem, which predicts the input instance's class. This prediction is made based on the majority vote of the k closest neighbours. The Equations (1)–(3) relevant to this process are also mentioned.

$$\text{Hamming distance:}\, d_{ij} = \sum_{k=1}^{p} \left| x_{ik} - x_{jk} \right| \tag{1}$$

$$\text{Manhattan distance:}\, d_1(p,q) = \|p-q\|_1 = \sum_{1=1}^{n} |p_i - q_i| \tag{2}$$

$$\text{Minkowski distance} = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \tag{3}$$

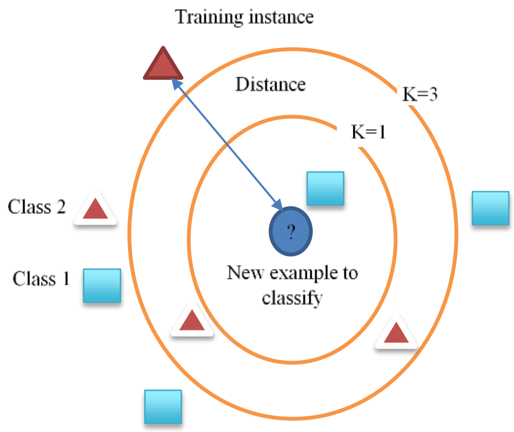

Figure 4. KNN example.

The Euclidean distance is typically the preferred approach for continuous variables. It is mathematically defined by the following equation:

$$\text{Euclidian distance} = \sqrt{\sum_{1=1}^{n}(q_i - p_i)^2\,; p \text{ and } q \text{ are the points in } n - \text{space}} \tag{4}$$

Euclidian distance works for similar characteristics. Use for many characteristics. The algorithm's ability to make accurate predictions is greatly influenced by the value of *k*. The *k* value is difficult to choose. Since every training set occurrence now has a greater weight in the decision process, smaller *k* values will impair accuracy, particularly in noisy datasets. It is suggested to choose k using the formula shown below Equation (5) as a general strategy.

$$k = \sqrt{n} \tag{5}$$

3.2.4. Support vector machines

SVM classification is another popular machine learning method[31]. Choose a hyperplane that optimally splits classes. "Support vectors" are the nearest points to the hyperplane that, if removed, would displace it. The support vector's distance from the hyperplane is called margin.

This is how the algorithm may be explained:

1) We define X and Y as the corresponding sets for the input and output. $(x_1, y_1), \ldots, (x_m, y_m)$ is the training set.
2) Given x, in order to anticipate y. The learning of the classifier opposition might be used to describe this issue $y = f(x, a)$, if the classification function's argument an is.
3) $F(x, a)$ may be taught by reducing the function that learns from training data's training error. R_{emp} is referred to as empirical risk at Equation (6), where *L* is the loss function.

$$R_{\text{emp}}(a) = \frac{1}{m}\sum_{i=1}^{m} l(f(x_i, a), y_i) = \text{Training error} \tag{6}$$

Additionally, we want to reduce the total risk. In this case, in Equation (7), *P*(*x*,*y*) is the joint distribution function of *x* and *y*.

$$R(a) = \int l(f(x,a),y)\mathrm{d}P(x,y) = \text{Test error} \tag{7}$$

The training error + Complexity term should be as low as possible. Consequently, select the group of hyperplanes in order to

$$f(x) = (w{\cdot}x) + b \tag{8}$$

$$\frac{1}{m}\sum_{i=1}^{m} l(w.x_i + b, y_i) + \|w\|^2 \text{subject to } \min_i |w.x_i| = 1 \tag{9}$$

Especially on “clean” datasets, SVMs are accurate. It also works well with high-dimensional datasets with more dimensions than samples. It’s better for huge datasets with plenty of noise or overlapping classes.

3.2.5. Random forest

Supervised classification is accomplished via random forest. Given that it is more like a collection of decision trees that may create a forest, random forest and decision tree have a direct relationship. The random forest model’s accuracy depends on tree count. More trees improve accuracy; it executes more slowly than one with a small number of trees. The structure of a randomly formed forest is explained in **Figure 5**.

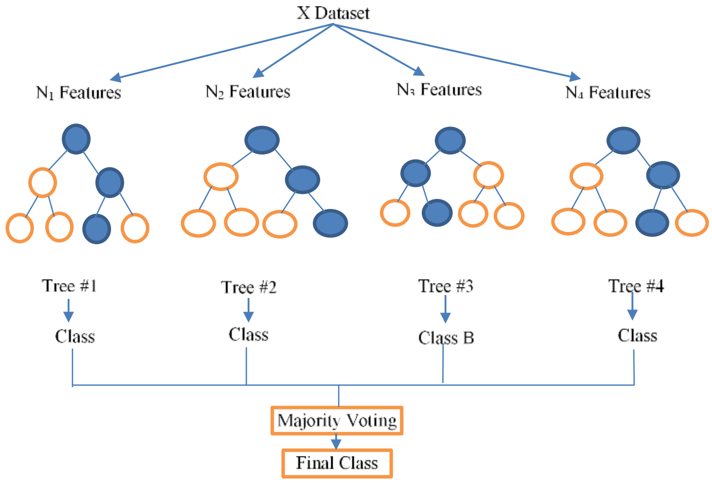

Figure 5. Random forest classifier.

3.2.6. Decision tree

Decision trees are often used in the construction of classification and regression models. A decision tree classifier is used in this study for the purpose of classifying malware [32]. It creates categorization models with a tree-like topology. The decision tree shown in **Figure 6** predicts the day’s weather.

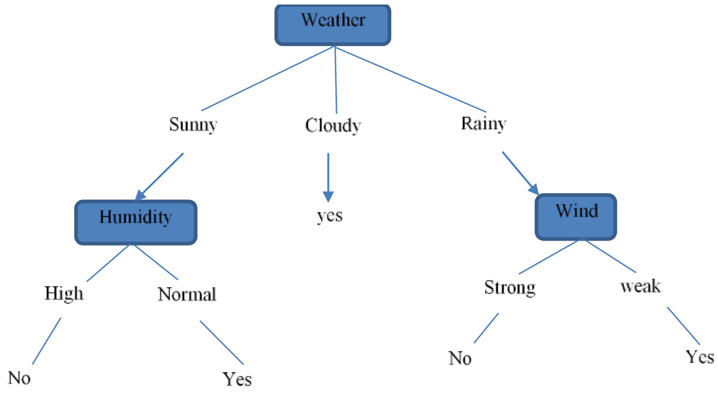

Figure 6. Decision tree.

3.2.7. Performance of machine learning algorithm

Cross validation is nothing but leave-one-out method. A new graph neural network model is presented based on a network traffic graph for malware detection is shown below. **Figure 7** represents the data flow diagram for machine learning algorithm

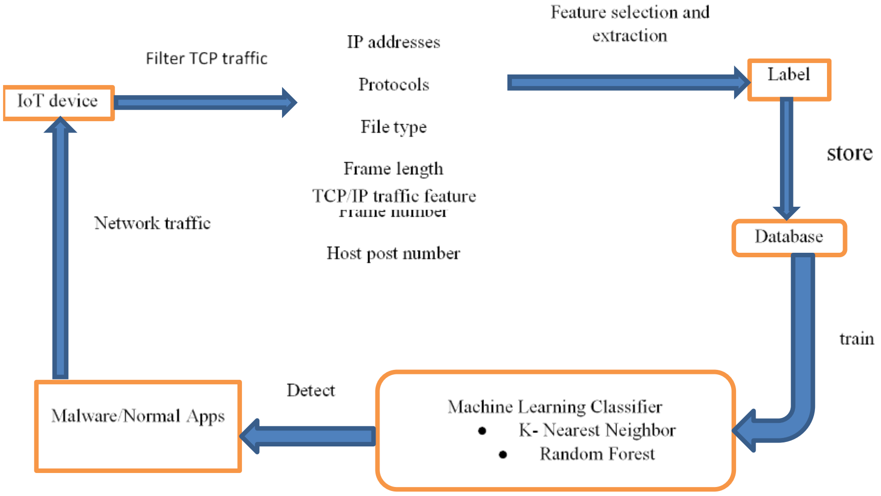

Figure 7. Data flow diagram for machine learning algorithm.

TCP packets are filtered, labelled, and stored in a database by the IoT device. The K-NN algorithm is utilized for the purpose of detecting ransomware in network traffic. The class that contains the most items among its K closest neighbors is used to categorize network traffic. To distinguish between various types of ransomware, the random forest classifier uses decision trees produced from tagged network data. According to tests in, the true positive rates of the random forest-based scheme and K-NN-based ransomware detection employing dataset are 93% and 69%, respectively[33]. The best offloading rate may be achieved by an IoT device using Q-learning in a ransomware detection method developed in[34] in the absence of knowledge about the target and the trace creation[35–37].

IOT devices’ network data set was gathered in pcap file format. The data sample that was converted and retrieved from the pcap file format utilizing wire shark is shown below.

Threats are classified based on the type of attack: Denial of service (DoS), This happens when a resource is overloaded and causes DoS to authorized users. To get around security, probes gather network information. Before the model is executed, the data set is divided into two sets: test and practice. The data is divided in the most demanding way for classifiers. The attack probability distributions in train and test data are not the same; as the test data set contains 16 of the 38 identified threats.

3.2.8. Results and discussion

With the use of experimental data, the usefulness of the recommended approach was also shown in this part, which covered the numerous datasets used in this study for testing the proposed network optimized classifier. In order to prove the effectiveness of intrusion detection, it concludes by demonstrating a comparison study. The datasets are first thoroughly discussed in as much detail as needed.

3.2.9. Dataset

The dataset selected and used in this paper is Aposemat IoT-23 (will be called IoT-23 from now onwards), by Avast AIC laboratory. Czech Republic is the country where the IoT-23 dataset was created and developed. The data was collected from the year 2018 to 2019. The dataset contains 20 malware captures and labels and rest are benign. Other than that, it carries 21 feature attributes. The dataset also carries pcap files, labelled (conn.log. Labelled) files. In this research, only labelled files were used as it was easier to work on with. The pcap files were not important in this project as they were difficult to handle and were mostly ignored throughout the project. The dataset was downloaded from the website called Stratosphereips.org. Two methods were available to download the dataset which was by downloading the whole folder in a compressed zip folder format or download the files separately, such as conn.log. Labelled and pcap. A total of 325,307,990 captures can be found in this dataset making the size of the dataset very large. The types of attack from the capture are shown in **Table 1**.

Table 1. Types of attack.

S. no.	Types of attack
1	Attack
2	Benign
3	C & C
4	C & C-FileDownload
5	C & C-Mirai
6	C & C-Torii
7	DDoS
8	C & C-HeartBeat
9	C & C-HeartBeat-Attack
10	C & C-HeartBeat-FileDownload
11	C & C-Part of A Horizontal Port Scan
12	Okiru
13	Okiru Attack
14	Part of a Horizontal Port Scan
15	Part of a Horizontal Port Scan Attack

3.2.10. Performance metrics

The following is a list of the four typical performance indicators for the identification of malware:

- True positive (TP): demonstrates that a harmful program has been successfully identified as malware.
- True negative (TN): demonstrates that a good ware application was successfully identified as a non-malicious application.
- False positive (FP): shows that a good ware has been incorrectly identified as a harmful one.
- False negative (FN): shows that a program is not considered harmful since malware has not been found.

Accuracy, recall, precision, F1-measure and area under the curve (AUC), which are all extensively used measures for measuring machine learning performance, were utilized to assess the efficacy of our suggested strategy (Equations (10)–(13)).

Accuracy:

The proportion of malware and genuine software packages to the total number of samples correctly identified by a classifier is as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{10}$$

Recall or detection rate:

As specified below, the percentage of ransomware samples that are accurately predicted is:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{11}$$

Precision:

A measure of the percentage of expected ransomware that is accurately classified as malware. So, the definition of precision is as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{12}$$

F1-measure score:

The result is a sum of accuracy and recall. It is sensible to choose to utilize the harmonic mean since they are both rates. This chart displays the F1 scoring formula: In light of this, the following is the formula for the F1 score:

$$\text{F1 score} = \frac{2PR}{P + R} \tag{13}$$

Area under the curve (AUC):

The AUC is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the model's performance at distinguishing between the positive and negative classes. AUC's value ranges from 0 to 1. AUC values of 0.0 and 1.0, respectively, reflect models with 100% inaccurate predictions and 100% correct predictions, respectively.

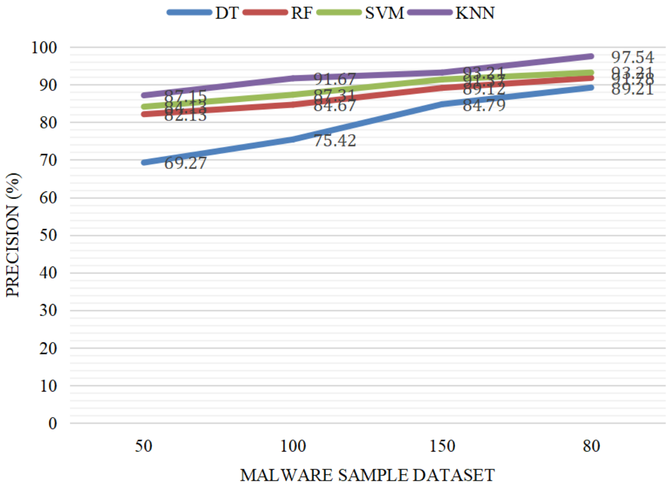

Figure 8. Precision results.

Figure 8 compares the precision results of other established approaches with the results of the proposed classifier. The proposed method gives higher precision results of 97.54%, whereas other methods such as DT, RF, SVM also gives higher precision for proposed KNN classifier based different methods such as 89.21%, 91.78%, 93.21% and respectively for Aposemat IoT-23 datasets. The suggested algorithm outperforms the current approaches in terms of results.

Figure 9 shows the comparison between the recall results of the network optimizer classifier and other well-known techniques. Recall results are higher with 98.12% using the suggested strategy, whereas other methods such as DT, RF, SVM also gives higher recall for proposed KNN classifier based different methods such as 86.12%, 89.21%, 93.4% and respectively for Aposemat IoT-23 datasets. In **Figure 9**, the proposed method yields better results than the current approaches, respectively.

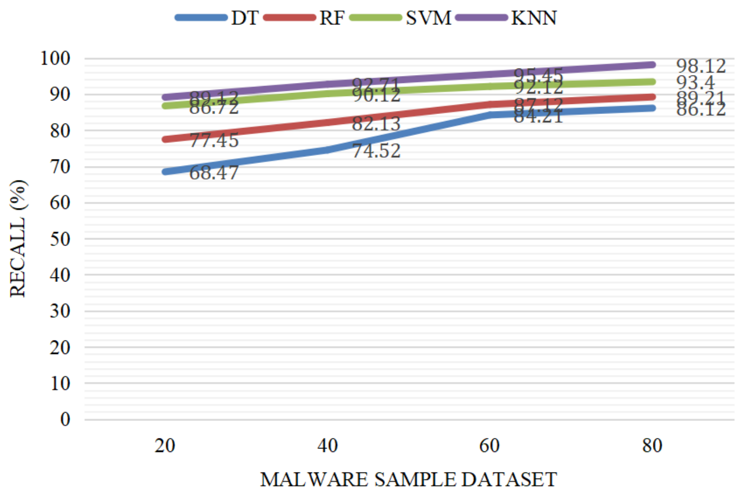

Figure 9. Recall results.

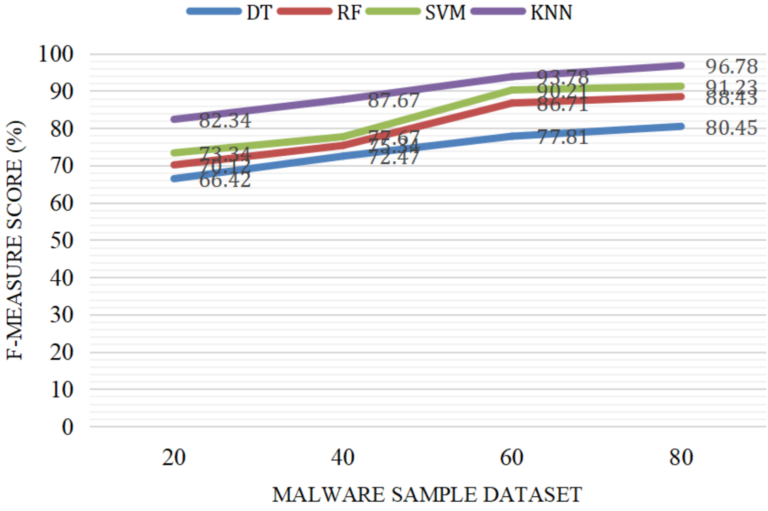

Figure 10. F-measure score results.

Figure 10 represents the F-measure value of the network optimizer classifier is compared with the other existing methods. The suggested approach produces result with a higher F-measure score of 96.78%, whereas other methods such as DT, RF, SVM also gives higher F-Measure score for proposed KNN classifier based different methods such as 80.45%, 88.43%, 91.23% and respectively for Aposemat IoT-23 datasets. In terms of results, the proposed algorithm performs better than existing methods.

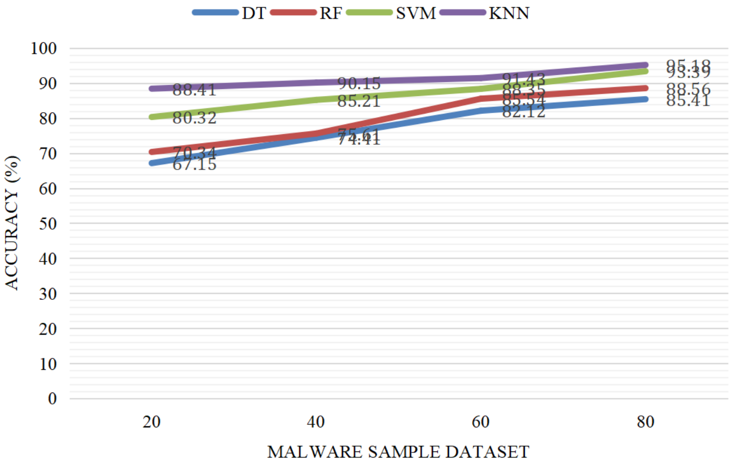

Figure 11. Accuracy results.

Figure 11 shows the comparison between the accuracy results of the Network Optimizer classifier and other well-known approaches. Compared to other approaches like DT, RF, and SVM, the suggested method gives higher accuracy results of 95.18% also gives higher precision for proposed KNN classifier based different methods such as 88.41%, 90.15%, 91.43% and respectively for Aposemat IoT-23 datasets. As shown in **Figure 11**, the proposed algorithm produces higher results than existing methods.

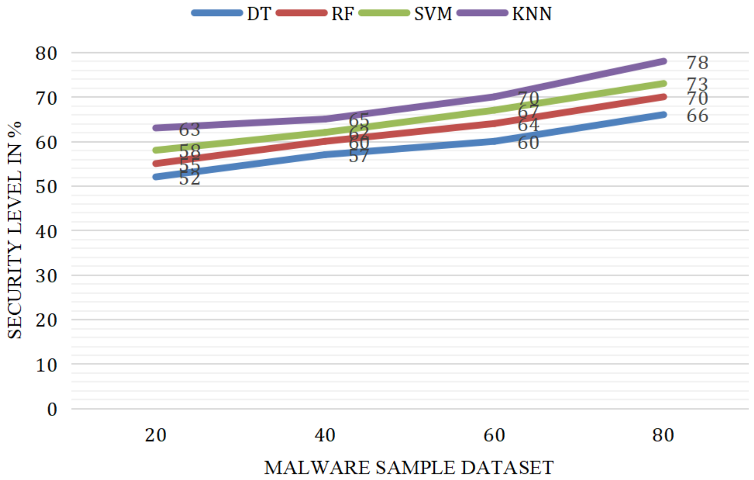

Figure 12. Security analysis for malware detection in NB-IoT.

The security attack detection using DT, RF, SVM, and KNN grounded routing is shown in **Figure 12**. The movement is altering the percentage (%) of encrypted security strategies for certain data. The graph clearly shows that, in response to attack time concerns, KNN grounded routing outperforms the other replicates with high attack detection rates. When compared to existing methods, the proposed method obtains high attack detection (78%), reaching 66%, 70%, and 73% correspondingly.

4. Conclusions

This article discusses a potential issue for the forthcoming NB-IoT network. The security of data broadcast and received via NB-IoT must be addressed as this industry grows and develops thanks to technical advancements. In this paper, security breaches in an NB-IoT topology are examined in depth, and the security issues at each level are described layer by layer. Particular attention is given to the crucial sectors, particularly defense security and security, since they need for strong data protection. The paper presented new method called network optimizer classification for detect the malware attacks in NB-IoT. The implementation of an NB-IoT network topology utilizing the network optimizer classification technique is proposed as a means to mitigate queuing delays (latency) at the BS. This approach has the potential to reduce energy consumption, extend battery lifetime, lower communication costs, improve network coverage, optimize bandwidth utilization, and enhance security by preventing malicious node attacks. The results show that a bot may efficiently fake authentic network traffic, and the overall security is compromised to 95%.

Future work

Certain limitations of federated learning, such as devices that crash in the midst of an operation, long upload and model update times, clients with little relevant data, and so forth, might lower the accuracy of the global model in subsequent operations. Future research must address these shortcomings, which make the global model far less accurate.

Author contributions

Conceptualization, RR and KK; methodology, KK; software, RR; validation, RR and KK; formal analysis, KK; investigation, RR; resources, KK; data curation, RR; writing—original draft preparation, RR; writing—review and editing, XX; visualization, KK; supervision, RR; project administration, KK. All authors have read and agreed to the published version of the manuscript.

Conflict of interest

The authors declare no conflict of interest.

References

1. Qiu M, Ming Z, Li J, et al. Phase-Change Memory Optimization for Green Cloud with Genetic Algorithm. IEEE Transactions on Computers. 2015; 64(12): 3528-3540. doi: 10.1109/tc.2015.2409857
2. Sfar AR, Chtourou Z, Challal Y. A systemic and cognitive vision for IoT security: A case study of military live simulation and security challenges. 2017 International Conference on Smart, Monitored and Controlled Cities (SM2C). doi: 10.1109/sm2c.2017.8071828
3. Anthi E, Williams L, Burnap P. Pulse: an adaptive intrusion detection for the internet of things. Living in the Internet of Things: Cybersecurity of the IoT—2018. doi: 10.1049/cp.2018.0035
4. Hou X, Li Y, Chen M, et al. Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures. IEEE Transactions on Vehicular Technology. 2016; 65(6): 3860-3873. doi: 10.1109/tvt.2016.2532863
5. Li Y, Zheng F, Chen M, et al. A unified control and optimization framework for dynamical service chaining in software-defined NFV system. IEEE Wireless Communications. 2015; 22(6): 15-23. doi: 10.1109/mwc.2015.7368820
6. Zayas AD, Merino P. The 3GPP NB-IoT system architecture for the Internet of Things. 2017 IEEE International Conference on Communications Workshops (ICC Workshops). doi: 10.1109/iccw.2017.7962670
7. Muteba F, Djouani K, Olwal TO, et al. Challenges and solutions of spectrum allocation in NB-IoT technology. Tshwane University of Technology. Pretoria, South Africa. pp. 2-7.
8. Chen M, Miao Y, Jian X, et al. Cognitive-LPWAN: Towards Intelligent Wireless Services in Hybrid Low Power Wide Area Networks. IEEE Transactions on Green Communications and Networking. 2019; 3(2): 409-417. doi: 10.1109/tgcn.2018.2873783
9. Moazzeni S, Sawan M, Cowan GER. An Ultra-Low-Power Energy-Efficient Dual-Mode Wake-Up Receiver. IEEE Transactions on Circuits and Systems I: Regular Papers. 2015; 62(2): 517-526. doi: 10.1109/tcsi.2014.2360336
10. Hoymann C, Astely D, Stattin M, et al. LTE release 14 outlook. IEEE Communications Magazine. 2016; 54(6): 44-49. doi: 10.1109/mcom.2016.7497765
11. Riaz S, Latif S, Usman SM, et al. Malware Detection in Internet of Things (IoT) Devices Using Deep Learning. Sensors. 2022; 22(23): 9305. doi: 10.3390/s22239305
12. Jeon J, Park JH, Jeong YS. Dynamic Analysis for IoT Malware Detection with Convolution Neural Network Model. IEEE Access. 2020; 8: 96899-96911. doi: 10.1109/access.2020.2995887
13. Asam M, Khan SH, Akbar A, et al. IoT malware detection architecture using a novel channel boosted and squeezed CNN. Scientific Reports. 2022; 12(1). doi: 10.1038/s41598-022-18936-9
14. Dartel B. Malware detection in IoT devices using Machine Learning [Bachelor's thesis]. University of Twente.
15. Mustafa Hilal A, Ben Haj Hassine S, Larabi-Marie-Sainte S, et al. Malware Detection Using Decision Tree Based SVM Classifier for IoT. Computers, Materials & Continua. 2022; 72(1): 713-726. doi: 10.32604/cmc.2022.024501
16. Pei X, Deng X, Tian S, et al. A knowledge transfer-based semi-supervised federated learning for IoT malware detection. IEEE Transactions on Dependable and Secure Computing. 20(3): pp. 2127-2143.
17. Tamás C, Papp D, Buttyán L, et al. SIMBIoTA: Similarity-based Malware Detection on IoT Devices. In: IoTBDS. pp. 58-69.
18. HaddadPajouh H, Dehghantanha A, Khayami R, et al. A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting. Future Generation Computer Systems. 2018; 85: 88-96. doi: 10.1016/j.future.2018.03.007
19. Mihoub A, Fredj OB, Cheikhrouhou O, et al. Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques. Computers & Electrical Engineering. 2022; 98: 107716. doi: 10.1016/j.compeleceng.2022.107716
20. Kaur A, Pal SK, Singh AP. Hybridization of K-Means and Firefly Algorithm for intrusion detection system. International Journal of System Assurance Engineering and Management. 2017; 9(4): 901-910. doi: 10.1007/s13198-017-0683-8
21. Sakr MM, Tawfeeq MA, El-Sisi AB, et al. An Efficiency Optimization for Network Intrusion Detection System. International Journal of Computer Network and Information Security. 2019; 11(10): 1-11. doi: 10.5815/ijcnis.2019.10.01
22. Abdelmoumin G, Rawat DB, Rahman A. On the Performance of Machine Learning Models for Anomaly-Based

Intelligent Intrusion Detection Systems for the Internet of Things. IEEE Internet of Things Journal. 2022; 9(6): 4280-4290. doi: 10.1109/jiot.2021.3103829

23. Srivastava A, Kumar A. A back propagation NN to optimize the IoT network. 2022 International Conference on Computer Communication and Informatics (ICCCI). Published online January 25, 2022. doi: 10.1109/iccci54379.2022.9740861
24. Azrour M, Mabrouki J, Guezzaz A, et al. Internet of Things Security: Challenges and Key Issues. Khan HU, ed. Security and Communication Networks. 2021; 2021: 1-11. doi: 10.1155/2021/5533843
25. Nižetić S, Šolić P, López-de-Ipiña González-de-Artaza D, et al. Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. Journal of Cleaner Production. 2020; 274: 122877. doi: 10.1016/j.jclepro.2020.122877
26. Weber M, Boban M. Security challenges of the internet of things. 2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). Published online May 2016. doi: 10.1109/mipro.2016.7522219
27. Hasan M, Islam MdM, Zarif MII, et al. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet of Things. 2019; 7: 100059. doi: 10.1016/j.iot.2019.100059
28. Sarker IH. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN Computer Science. 2021; 2(3). doi: 10.1007/s42979-021-00592-x
29. Soofi AA, Awan A. Classification Techniques in Machine Learning: Applications and Issues. Journal of Basic & Applied Sciences. 2017; 13: 459-465. doi: 10.6000/1927-5129.2017.13.76
30. Asharf J, Moustafa N, Khurshid H, et al. A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions. Electronics. 2020; 9(7): 1177. doi: 10.3390/electronics9071177
31. Ioannou C, Vassiliou V. Network Attack Classification in IoT Using Support Vector Machines. Journal of Sensor and Actuator Networks. 2021; 10(3): 58. doi: 10.3390/jsan10030058
32. Hemanth DJ. Improved Malware Detection for IoT Devices Using Random Forest Algorithm Comparing with Decision Tree Algorithm. pp. 597-603.
33. Narudin FA, Feizollah A, Anuar NB, et al. Evaluation of machine learning classifiers for mobile malware detection. Soft Computing. 2014; 20(1): 343-357. doi: 10.1007/s00500-014-1511-6
34. Xiao L, Li Y, Huang X, et al. Cloud-Based Malware Detection Game for Mobile Devices with Offloading. IEEE Transactions on Mobile Computing. 2017; 16(10): 2742-2750. doi: 10.1109/tmc.2017.2687918
35. Razzak I, Moustafa N, Mumtaz S, et al. One-class tensor machine with randomized projection for large-scale anomaly detection in high-dimensional and noisy data. International Journal of Intelligent Systems. 2021; 37(8): 4515-4536. doi: 10.1002/int.22729
36. Haider SK, Jiang A, Jamshed MA, et al. Performance Enhancement in P300 ERP Single Trial by Machine Learning Adaptive Denoising Mechanism. IEEE Networking Letters. 2019; 1(1): 26-29. doi: 10.1109/lnet.2018.2883859
37. Satpathy SK, Vibhu V, Behera BK, et al. Analysis of Quantum Machine Learning Algorithms in Noisy Channels for Classification Tasks in the IoT Extreme Environment. IEEE Internet of Things Journal. 2024; 11(3): 3840-3852. doi: 10.1109/jiot.2023.3300577