Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 18 No. 19 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijim.v18i19.50315

PAPER

Enhanced Machine Learning Based Network Traffic Detection Model for IoT Network

Mazen Alzyoud[1](✉), Najah Al-shanableh[1], Eman Nashnush[2], Rabah Shboul[1], Raed Alazaidah[3], Ghassan Samara[3], Safaa Alhusban[1]

[1]Computer Science Department, Al al-Bayt University, Mafraq, Jordan

[2]Computer Science Department, University of Salford, Salford, United Kingdom

[3]Computer Science Department, Zarqa University, Zarqa, Jordan

malzyoud@aabu.edu.jo

ABSTRACT

Ensuring the security of networks is a significant hurdle in the rollout of the Internet of Things (IoT). A widely used protocol in the IoT ecosystem is message queuing telemetry transport (MQTT), which is based on the published-subscribe model. IoT manufacturers are expected to expand their usage of the MQTT protocol, which is expected to increase the number of cyber security threats against the protocol. IoT settings are crucial to overcoming scalability and computing resource issues and minimizing the characteristics needed for categorization. Machine learning (ML) is extensively used in traffic categorization and intrusion detection. This study proposes a ML-based network traffic detection model (MLNTDM) to enhance IoT application layer attack detection. The proposed architecture for the MQTT protocol is evaluated based on its effectiveness in detecting malicious attacks and how these affect various MQTT brokers. This study focuses on low-power-consuming ML algorithms for detecting IoT botnet offenses and identifying typical attacks and their responses. With this framework, each network flow provides information that can help identify the source of generated traffic and network assaults. Results from our approach, as shown in the experiment, prove more accuracy.

KEYWORDS

machine learning (ML), Internet of Things (IoT), network traffic detection, message queuing telemetry transport protocol (MQTT), traffic classification, computing resource, scalability

1 INTRODUCTION

Ensuring the security and legitimacy of network communication is of paramount significance due to the exponential growth in internet users [1]. The old methods of monitoring network traffic seldom succeed when effectively identifying and controlling emerging cyber risks [2]. This has led to an explosion in the development of advanced machine learning (ML) models tailored for identifying network traffic [3]. An enhanced ML-based model for detecting network traffic is introduced in

Alzyoud, M., Al-shanableh, N., Nashnush, E., Shboul, R., Alazaidah, R., Samara, G., Alhusban, S. (2024). Enhanced Machine Learning Based Network Traffic Detection Model for IoT Network. *International Journal of Interactive Mobile Technologies (iJIM)*, 18(19), pp. 182–198. https://doi.org/10.3991/ijim.v18i19.50315

Article submitted 2024-05-29. Revision uploaded 2024-06-20. Final acceptance 2024-07-19.

this study [4]. Cyber risks in today's interconnected world are always evolving, and this approach was created to tackle such challenges [5]. Compared to traditional procedures, this methodology delivers notable advantages in efficiency, accuracy, and adaptability [6]. It uses advanced techniques and approaches in ML [7]. The primary goal of this study is to enhance the skills of network administrators and security analysts to better detect and respond to suspicious activity inside network traffic. The proposed approach can proactively safeguard systems from emerging cyber dangers because it can learn and adapt to changing patterns of benign and malicious behavior. Because of this, the model may now use ML capabilities [8]. The model's ability to classify different types of network abnormalities correctly, analyze massive amounts of real-time network traffic data, and provide actionable insights to help with incident response speed are some of its key characteristics. The model's scalability and configurability make it easy to incorporate into current network architectures in various industries and enterprise settings. Finally, strong privacy-preserving procedures should be implemented to protect sensitive information because there are privacy risks with collecting and analyzing data from network traffic [9]. To fully utilize improved ML-based network traffic detection models for efficiently protecting Internet of Things (IoT) networks, it is vital to address these problems [10].

Supervised, unsupervised, and semi-supervised mastering are current techniques used in models for stepped-forward machine learning-based total network traffic detection in IoT networks [11]. Models skilled in the use of supervised mastering techniques, together with support vector machines (SVMs) and deep neural networks (DNNs), may additionally distinguish between regular and ordinary community traffic [12]. Anomaly detection using unsupervised learning methods, including autoencoders and clustering algorithms, can be completed without labeled records by recognizing outliers in everyday visitor patterns [13]. By incorporating features of each method, semi-supervised gaining knowledge improves version accuracy using a smaller set of labeled facts alongside a bigger set of unlabeled records [14]. Ensemble mastering strategies combine diverse models to enhance detection accuracy; examples encompass gradient boosting and random forests [15]. Problems with efficiently implementing these techniques remain notwithstanding those advances [16]. It remains a large assignment to teach ML fashions efficaciously and scalable on IoT devices with constrained sources; this demands optimization techniques to reduce computational overhead [17]. In addition, IoT settings are always changing and diverse, which makes it tough to model and discover anomalies efficiently. In addition, keeping users' confidence and regulatory compliance in mind demands resolving privacy reservations related to collecting and analyzing touchy records from community traffic. Successfully shielding IoT networks is based on overcoming those problems and using network companies' detection methods based on advanced device studying.

The main contribution of the paper are as follows:

- This study presents a machine learning-based network traffic detection model (MLNTDM) for detecting network traffic to fix security weaknesses using the MQTT protocol widely used within IoT ecosystems.
- The significance is identified by this model's use of low-power-consuming ML techniques towards addressing scalability issues inherent to IoT contexts where resource limitations exist.
- The experimental results show that the MLNTDM model effectively identifies the origins of valid and destructive operations by collecting data for every network flow and improves IoT network security by detecting and responding to application layer attacks.

The paper continues as follows: The literature review in Section II provides an improved model for detecting network traffic in IoT networks using ML. Mathematical details of the proposed MLNTDM model for detecting network traffic based on ML are given in Section III. The results, analysis, and comparisons with previous works are exhibited in Section IV of the experiment. A summary of the findings is presented in Section V.

2 LITERATURE REVIEW

Protecting networks from ever-changing threats is of utmost importance in the IoT domain. Several academics have dedicated themselves to creating cutting-edge methods for protecting IoT ecosystems from harmful actors. Study projects use various ML techniques, such as convolutional neural networks (CNNs) and ensemble methods, to better detect risks and unusual behavior in the Internet of Things.

Q. Abu Al-Haija et al., [18] propose developing and evaluating machine-learning-based Darknet traffic detection systems (DTDS) for IoT networks, employing six supervised machine-learning techniques. Evaluation of the CIC-Darknet-2020 dataset shows bagging decision tree ensembles (BAG-DT) achieve 99.50% classification accuracy with low inferencing overhead, outperforming other techniques and improving previous state-of-the-art models by 1.9~27%.

Using ML at the network edge (ML-NE), Salman et al. [19] present a method for identifying IoT devices and detecting malicious traffic. Features are extracted per network flow to detect threats, identify device types, and classify traffic. The most effective method is random forest. A device-type identification accuracy of 94.5%, a traffic-type classification accuracy of 93.5%, and an abnormal traffic detection accuracy of 97.5%.

Using characteristics retrieved from network traffic, R. Kumar et al. [20] perform a thorough comparative study of ML approaches (CA-MLA) for IoT traffic classification. By analyzing a publicly available dataset of 20 days' worth of data from 20 IoT devices, we can identify important features and compare multiple state-of-the-art ML algorithms according to their classification accuracy, training time, and speed. This will help us choose the best method for different situations.

M. Shafiq et al. propose a novel method for accurate malicious traffic detection (AMTD) in IoT networks. [21] We provide CorrAUC, a new feature selection metric, and show how to perform efficient feature filtering using it. The author combined Shannon entropy and TOPSIS for feature validation. Four ML methods are evaluated on the Bot-IoT dataset, and the results show an average efficiency of more than 96%.

Saba et al. describe a CNN method based on anomalies for intrusion detection in IoT networks. [22] The model uses deep learning to analyze IoT traffic efficiently and spot intrusions and unusual behavior. Its 99.51% and 92.85% accuracy percentages, respectively, when trained and tested on the NID and BoT-IoT datasets, show how successful it has been in enhancing IoT security.

Raymundo Buenrostro-Mariscal et al. [23] suggested deep learning applications for the next generation of cognitive networks. To provide a benchmark for future endeavors in this area, this study provides an overview of the present state of the art concerning deep learning in applications for intelligent cognitive networks. Articles that addressed the difficulties of existing cognitive networks and offered solutions based on deep learning were considered after a comprehensive literature search across three databases. This led to the examination of fourteen articles. The findings demonstrated that many viewpoints and experimental approaches have been taken in recent years to examine the technical viability of using deep learning algorithms

for optimizing cognitive data networks. Its potential effects on resolving some of the most basic problems with today's wireless networks are also covered.

These findings suggest that MLNTDMs are superior to traditional approaches in fortifying IoT security, providing proactive and robust threat mitigation strategies for IoT environments.

3 PROPOSED METHOD

An extensive network of interconnected embedded devices that can exchange data and instructions in "smart" environments is known as the IoT. Proper operation and identification of harmful activities need constant monitoring of IoT network traffic produced by IoT devices. Network traffic classification of IoT devices are one such critical operation. If the administrator desires to ensure that Quality of Service is being properly implemented or identify any malicious IoT devices, they may do so by monitoring the actions of these devices. Many different ML techniques have been suggested in the literature as potential solutions for the problem of IoT traffic classification. These ML algorithms rely on data collected from the IoT devices, features retrieved from network traffic, the location of the IoT deployment, and other factors to determine their accuracy. And since it is a manual process, feature selection and ML algorithm development are susceptible to human errors. For this reason, appropriate ML techniques and study into network traffic characteristics are crucial for the efficient and correct classification of IoT data. Here, this study looks at several renowned ML methods and compares them using a variety of useful data retrieved from IoT network traffic. It starts with processing the network traces to pull out the important elements. After that, this study examined the most current surveys on IoT traffic classification to choose advanced ML methods. This study then compared the ML algorithms' efficiency, speed, training time, classification accuracy, and other metrics. With more devices being connected, network security is becoming increasingly important. The MQTT protocol, necessary for IoT communication, is now vulnerable to cyberattacks. The study presents an MLNTDM (enhanced ML-based networking traffic monitoring model) for IoT settings. For recognizing IoT botnet attacks, the proposed model focuses on power-efficient algorithms that can identify and detect them in a scalable way. This study illustrates that MLNTDM enhances network security by detecting and responding to multiple MQTT-based attacks.

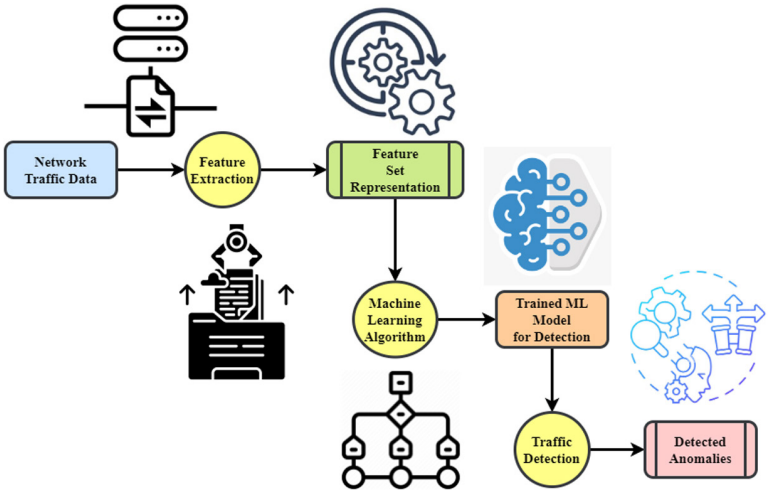

Fig. 1. Gateway for handling data on network connections

The first step in processing this data involves collecting initial information about network activity. Subsequently, this data undergoes feature extraction to create a feature set representation; hence, it's looked at closely to identify and extract important traits. Figure 1 shows some of the captured network traffic features, such as packet size, method, and source/destination address. These features extracted from the collected datasets are then passed through an ML algorithm during the next stage. An anomaly-alert-trained ML is built with patterns and correlations learned from the data by an ML algorithm. This study uses this model to differentiate abnormal patterns concerning network behavior from typical ones. The model is used to identify traffic after it has been trained. Confirmation of the absence of any deviation from anticipated fashion instantly follows receipt of internet connection acquired information immediately upon receipt. Any questionable activities are documented and marked for future study or corrective action. In conclusion, by providing rapid automatic anomaly detection, this channel seeks to maintain the safety and integrity of networks. The system can respond to new threats and protect the network architecture from harm by using ML to learn.

$$y_{ypk} = \frac{y - \min(y)}{\max(y) - \min(y)} \times (c - b) + b, t = Y + \forall (b - Y) \tag{1}$$

Equation 1 illustrates how the result y of the MLNTDM model is normalized (y_{ypk}); the given equation is consistent with the suggested techniques $\frac{y - \min(y)}{\max(y) - \min(y)}$. To make comparisons and interpretations easier $\forall$, the output values are normalized within a range set by variables (c) and (b).

$$\nabla = \frac{1}{n} \sum_{j=2}^{n} Y_j + Y_D \sum_{j=1}^{p} A_z - \Delta + \frac{1}{p} (Y_d^U \times Z_D) \tag{2}$$

Characteristics of the network's traffic (A_Z), Y_j (individual network traffic data), Y_D (a constant suggesting an initial measure), and Δ (a threshold parameter) were all given by equation 2. The equation measures the general efficacy of the network in detecting anomalies by averaging the total of these variables over the dataset (n). Adding Y_d^U and Z_D to the assessment procedure makes the model more flexible to different network circumstances.

$$D \times P = U \nabla P^Q + [p_1, p_2, \ldots p_l] + Y_{dsw} = Y_d \times C \tag{3}$$

It includes things D, which stands for network dynamics, P, which stands for parameters impacting network behavior, U which indicates the connection between parameters Q, which is a coefficient that tells how different parameters affect performance, and p_1 which are particular variables that affect network performance. Incorporating Y_{dsw} and C into the model makes it more sensitive to changing network circumstances by adjusting the output (Y_d) depending on contextual cues (∇). Given the wide variety of compression algorithms used in networking and data transmission, it is crucial to be familiar with the precise context or whole name of the protocol in question while discussing COMP Protocol. ML algorithms may find it simpler to evaluate and spot abnormalities if MQTT's dependable message delivery leads to more predictable traffic patterns. On the other hand, the reduced overhead of Constrained Application Protocol (CoAP) may cause traffic patterns to be more unpredictable. CoAP is more suited to applications with less vital data, where the odd packet loss is tolerable, but MQTT is better suited for high-frequency, critical

data owing to its dependable delivery. Although the detection approach may have limitations when processing complicated traffic patterns, the increased device deployment capabilities made possible using CoAP make it an appropriate option for extremely limited contexts. The strong security features of MQTT could be more useful if the detection model is highly dependent on data integrity and confidentiality.

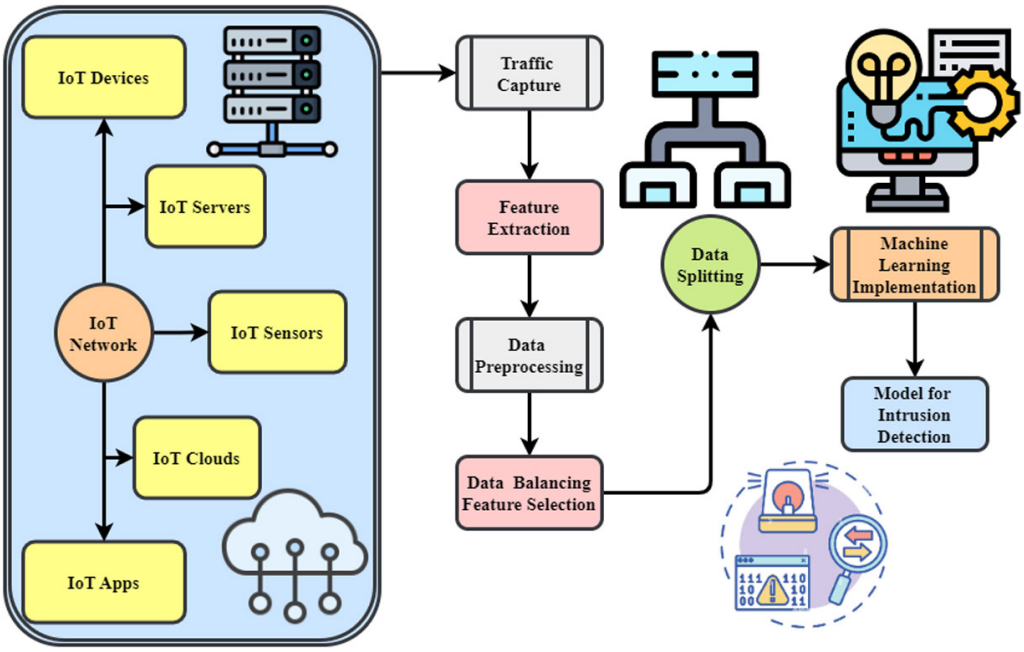

Fig. 2. Protocol for the IoT network's future intrusion detection system on machine learning

Using ML techniques, the figure depicts the process of the suggested intrusion detection system specifically designed for IoT networks. Several parts of the framework make intrusion detection strong and effective, as shown in Figure 2. Pre-processing procedures include cleaning, normalizing, and getting ready to analyze raw data from IoT networks. This stage aims to ensure the data is ready for processing. Extraction is carried out after pre-processing to find useful characteristics in the data collected from the IoT network. These network communication elements capture characteristics, including packet headers, cargo content, and communication patterns.

After retrieving characteristics, the following step is to train the model using ML methods. It provides and tests three ML models to detect intrusions, each optimized for data collected from IoT networks. After that, the mathematical models that have been trained are put into action to identify intrusions instantly. Based on learned tendencies and abnormalities, algorithms analyze incoming network data and label it harmless or malicious. The network is tested extensively to see how well it can identify and categorize intrusions. The suggested framework provides an all-encompassing method for detecting intrusions in the IoT, strengthening network security, and efficiently mitigating any dangers through ML approaches.

$$i_1 \times i_j = SeMP^2 - (X_j \times i_{j-1} + C_j) + softmax\,(X_j + Y_k) \tag{4}$$

Terms i_1 and i_j denote the inputs and the relationships in equation 4. Functions that indicate the extraction of features or modifications inside the model are denoted by the name *SeMP*. A constant C_j and two variables X_j and Y_k that affect the framework's

decision-making procedure are included. A probabilistic output, maybe utilized for task classification inside the MLNTDM, is indicated by the existence of *softmax*.

$$M2_{sew} = 0.5 \times m2_{mkl} \times \left(\lceil X_1 \rceil^2 + |X_2|^2 + \cdots + |X_p|^2 \right) \quad (5)$$

An adjusted version of an attribute metric ($M2$) for network activity analysis is represented by equation 5 as $M2_{sew}$. It takes the squared norms of the variables entered (X_1 to X_p) and is generated from $m2_{mkl}$, which may be an existing measurement or model.

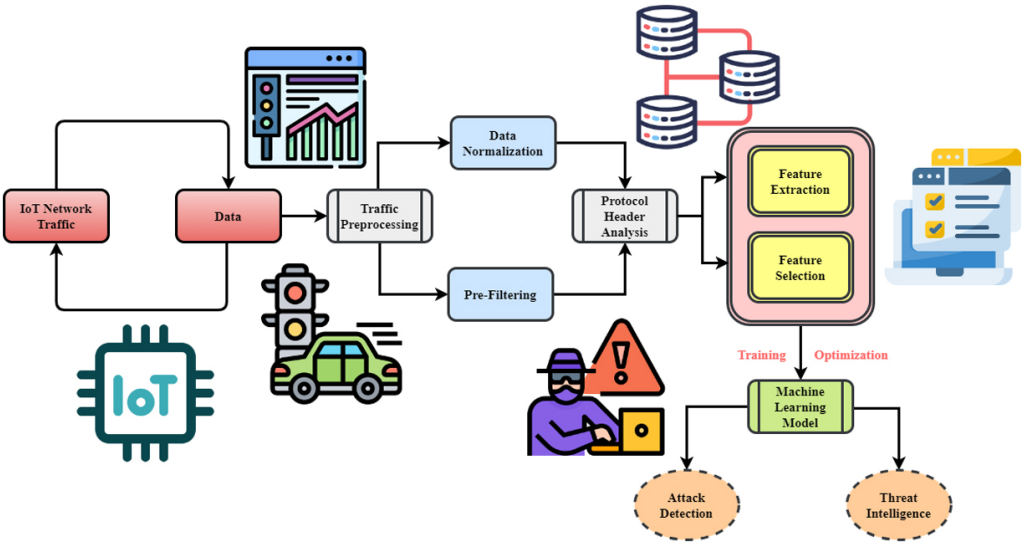

Fig. 3. Machine learning-based network traffic detection

For IoT systems in particular, Figure 3 lays out a thorough architecture for traffic redirection using ML. IoT network traffic, which includes various data streams from linked devices, enters the system first. Data normalization and pre-filtering form part one of traffic pre-processing, whose sole aim is to maintaining accuracy and dependability in the figures being given out. Then, these messages should be sent to the network for scrutiny of the structure and content via protocol header examination and decoding of network packets. The next step involves extracting specific attributes from processed data to understand network traffic properties. These features will be used to train a machine-learning model. At its core, this template is designed for training artificial intelligence models. At this stage, complex algorithms show how to use attributes gathered to identify patterns and abnormal activities that may suggest potential threats in the network. Afterwards, this model is thoroughly tested during training and validation to determine whether it can accurately detect or reroute malicious traffic. In addition, the model operates effectively when switched on for real-time attack detection and threat intelligence collection purposes. It continuously monitors the IoT networks' traffic and identifies, detects, or diverts potentially dangerous behaviors.

$$g_u = \forall(X_g \times [i_{u-1}, y_u] + c_k) \times \nabla(Z_j \times |i_{k-1}|) \quad (6)$$

Equation 6, in which a constant (c_k) is used to regulate the interaction between the input parameters (X_g) and prior model results (i_{u-1} and y_u), with g_u representing

the function's output. The slope function indicates a change in direction, and the Z_j is a further parameter that affects the way the model makes decisions.

$$j_u = \exists \cot j \times (X_d \times [j_{p-1}, y_u] + d_f) \times (\sec \nabla + E_g) \tag{7}$$

In this equation 7, X_d, y_u, and d_f are input variables, and the derived value j_u is affected by the interplay of these variables with the preceding model outputs. Equation 7 evaluates network traffic patterns. The cotangent and secant functions may represent alterations to the input data or the model's results sec. An outside factor appears to impact the model's decision-making procedure due to its inclusion E_g.

$$D_u = \aleph(\beth^{\gamma} - 1 + (\delta + \alpha) + C_1) + \mu_w(1 + \varphi) \tag{8}$$

It is in harmony with values of D_u are affected by the parameters $\aleph$, γ, δ, and α, which are probably properties and constants of the network. The addition of indicates a starting point or offset element C_1, and a weighting coefficient is denoted by μ_w. The use of $\aleph$ and φ suggests that the input information or model outputs have been subjected to nonlinear adjustments or modifications.

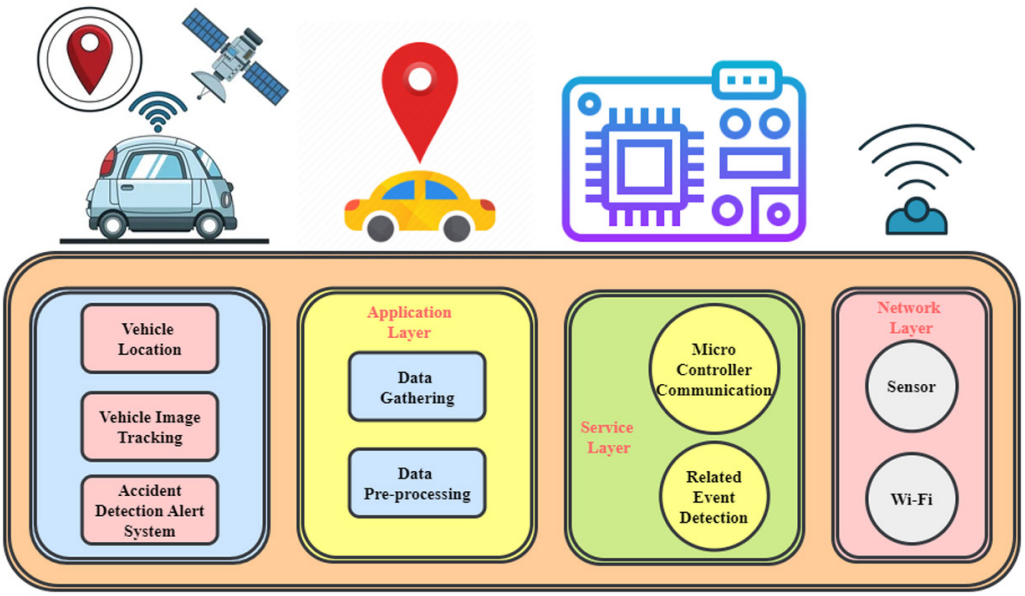

Fig. 4. A secure (IoT) solution for traffic control based on machine learning

The suggested approach combines adaptive traffic management with an accident alarm sound system and secures advanced transport-related event recognition to create a safe and smart transportation system. Each of the four pillars of this all-encompassing strategy helps build a more modern and secure transportation network. The application layer must continuously monitor car position and image tracking to detect and respond to incidents. By immediately signaling incidents or crises, the accident alarm sound system improves safety and allows for quick response. As the name suggests, the service layer collects and processes data from different sources. Figure 4 shows how this layer organizes and optimizes data for further analysis and transmission. Event recognition for advanced traffic security facilitates effective communication of vehicle information on the network. This layer uses strong encryption and authentication algorithms to protect sensitive data,

ensuring it stays private and uncompromised while being sent. The sensing layer uses the sensors to collect real-time data on traffic, vehicle behavior, and surroundings. These sensors provide the raw data for precise analysis and decision-making across the transport network. Figure 4 shows the method's design framework, which aims to build a durable, intelligent, and trustworthy transportation system by connecting and using each layer.

$$T_{\Delta}(u^{+}) = \begin{cases} t_{\beta}(u) - 1, & if\ t_{\beta}(l) < 0 \\ 0, & if\ t_{\beta}(l) = 0 \end{cases} \tag{9}$$

Equation 9 showcases the accuracy analysis of how the MLNTDM model handles certain cases where $T_{\Delta}(u^{+})$ stands for an output value that is decided by assessing the outcome of a different function $t_{\beta}(u)$, which could suggest a thresholding process. The given condition makes it clear that the result is modified appropriately if the input of $t_{\beta}(l)$, drops below zero.

$$\nabla_{p} = \ni_{p} S_{m} + f_{g}^{+} + \bigcup_{n=1}^{m} (W_{p} \cap X_{p}) \times \Lambda_{(\Delta+\nu)}^{(\forall+1)} \partial\nabla \tag{10}$$

The detection time analysis ∇_{p} is denoted by the variables S_{m}, f_{g}^{+}, W_{p}, and X_{p}, which are probably parameters for the network and the model, in equation 10. A logical operator denotes an association 1 or crossing operation represented by the symbol ∀. Perhaps indicating that the model is sensitive to shifts to network properties, the presence of ∂∇ implies a partial derivative function ν.

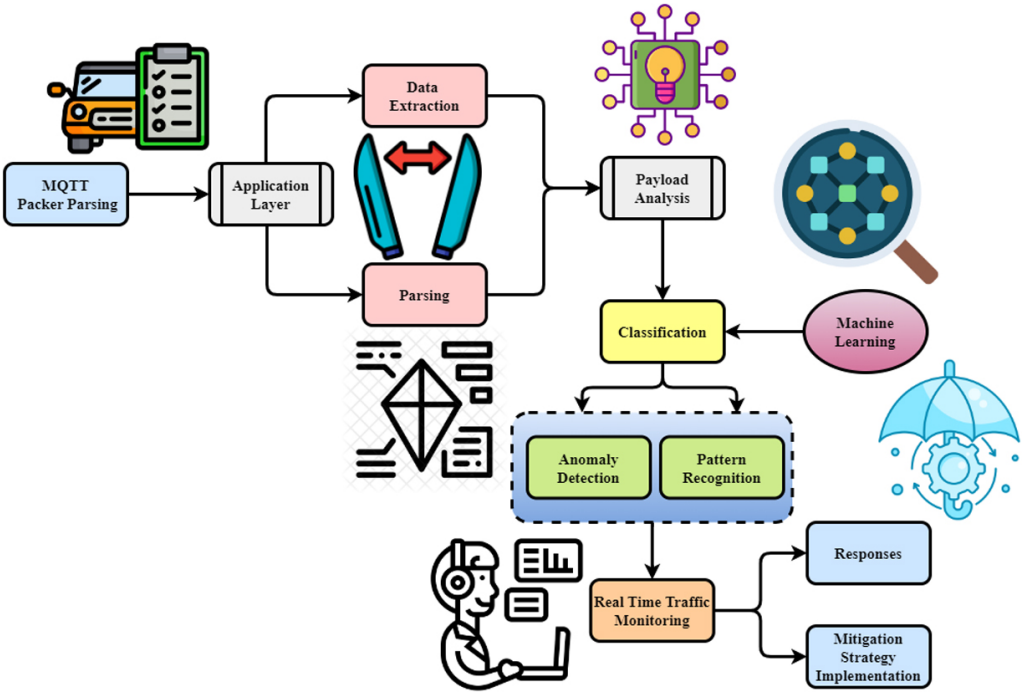

Fig. 5. MQTT packet parsing

Figure 5 presents a detailed architecture of traffic redirection using ML specifically for IoT networks. This includes different data streams coming from connected devices in an IoT network. Traffic preprocessing consists of data normalization

and pre-filtering to ensure the soundness and reliability of the data. Inspection and decoding of protocol headers examine the structure of network messages and their content. Accordingly, the subsequent step is to extract and select essential features from the pre-processed information that capture the fundamental properties of the traffic in a network. Following this, such characteristics are used to train the ML framework, as shown in Figure 5. On the other hand, this is where training and optimizing ML models are at their core. In this stage, sophisticated algorithms are taught to use the characteristics obtained to detect patterns and outliers that may indicate possible network threats. The last step involves extensive testing of the model, upon which evaluations will be made to determine if it can accurately identify and redirect harmful traffic. The deployed model then identifies real-time attacks while collecting threat intelligence. Due to this fact, constantly monitoring network activities reinforces the IoT's security posture by identifying and redirecting possibly harmful actions. By implementing these methods within a comprehensive platform, managers responsible for IoT networks would actively mitigate cyber risks, block dangerous patterns, and ensure their backbone remains intact.

$$\beth(\mu) = \beta_p + \rho_q \times J_k + Q(Y \mid D = d) \tag{11}$$

This Equation 11 describes the scalability analysis with $\beth(\mu)$ a calculated value that is affected by variables β_p, ρ_q, J_k, and $Q(Y \mid D = d)$, which probably reflect model coefficients and represent metrics for the network. The model considers data dynamics, as indicated by the conditional probability expression.

$$Q(Y \mid D = d) = Q(Y_2 \mid D = d) \times Q(Y_2 \mid D = d) \tag{12}$$

The resource consumption analysis for equation 12 shows the conditional likelihood of events under particular conditions, $D = d$, and the chance of seeing the event Y_2 is denoted as $Q(Y_2 \mid D = d)$. The fact that this change is determined by multiplying two equal conditional odds in the equation implies that the model's analysis is repeated or recursion.

$$E = (Y_1, z_1),(Y_2, z_2), \ldots, (Y_p, z_q) + Y_{test} = \arg\cos(F_j) = j \tag{13}$$

The robustness analysis is determined on E is a dataset that includes pairs of observable traffic on the network characteristics (Y_1) and related labels or classifications (z_1), as shown in the preceding equation 13. The data set used to evaluate the model's performance is denoted by Y_{test}. For each feature vector F_j, where j is the index of the most comparable feature vector, the cosine relationship to the dataset may be calculated using the function $\arg\cos(F_j) = j$. The smart thermostat in a smart home has an unexpected spike in traffic around 3 AM, when it usually has minimal communication, according to the detection model based on ML. In the payloads, you'll find instructions to crank up the thermostat to an absurd level. In response to what it interprets as an attempted intrusion, the model notifies the homeowner and temporarily removes their ability to control the thermostat remotely. This detection and mitigation strategy provides a strong answer to the problem of protecting IoT environments by centring on the MQTT protocol. It can detect and react to any cyberattacks by protecting the security and functionality of smart home technologies. With this method, this study can see how machine learning-based network traffic identification works in IoT networks in real-time and how useful it is.

For this reason, when protecting IoT networks, especially those operating under the MQTT protocol, improved ML-based approaches presented by this study offer a considerable solution. ML algorithms tailored for low power consumption make MLNTDM more accurate in detecting malicious behaviors, improving safety in IoT environments. The proposed model has been evaluated through experiments that show promising results about differentiating between benign and malicious communication, which might improve security measures for IoTs. Further improvements can be added to harden such platforms against ever-changing cyberattacks.

4 RESULTS AND DISCUSSION

A comprehensive evaluation of the proposed MLNTDM's efficacy and dependability in detecting and reducing cyber threats in IoT networks is carried out via accuracy, detection time, scalability, resource consumption, and resilience analyses.

Dataset description: This dataset on network traffic gives comprehensive information on the digital exchanges that occur within a network setting. On October 9th, 2023, at the University of Cincinnati in Ohio, a Kali machine was utilized to record 394,137 occurrences using Wireshark. The data for these cases was collected over one hour (https://www.kaggle.com/datasets/ravikumargattu/network-traffic-dataset) [23]. Each of the seven dataset attributes describes a different crucial aspect, such as the IP addresses of the packets' origin and destination, the protocols used, and their lengths. This dataset contains numerical, nominal, and temporal data that may be used for ML tasks, such as detecting network intrusions, classifying traffic, and identifying anomalies. Among these uses are traffic categorization and network intrusion detection.

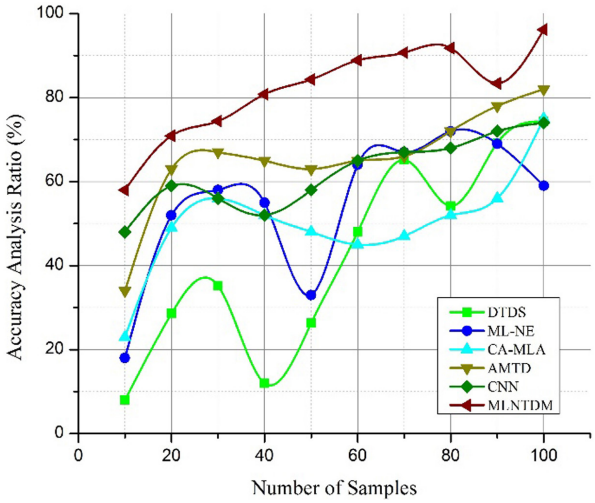

Fig. 6. Accuracy analysis

In Figure 6, the effectiveness of the MLNTDM in identifying and mitigating cyber risks for IoT networks ought to be evaluated well through an accuracy take and explore. The effectiveness of the MLNTDM in identifying suspicious or malicious behavior in IoT network statistics can be thoroughly tested by engaging in large exams and evaluations, producing 98.7%. The accuracy with which the model can distinguish between traditional and out-of-the-ordinary network interest is investigated by measuring some parameters: the actual effective rate, false advantageous

rate, accuracy, and F1-score. The accuracy investigation additionally consists of trying out the version thoroughly below various attack vectors and situations to peer how properly it holds up in diverse IoT settings. The accuracy study compares the version to baseline methods and other detection mechanisms; viewing it will give a good idea of how much better the MLNTDM is at detecting things. Future upgrades and modifications to the MLNTDM may be guided using sensitivity analysis, which assists in identifying ability boundaries and improvement regions. A radical accuracy is a crucial yardstick for determining whether or not the MLNTDM is effective and dependable in shielding IoT networks from cyber assaults, enabling the construction of extra steady and resilient IoT infrastructures. The high accuracy is attributed to the advanced ML algorithms that leverage MQTT-specific features. This allows the model to differentiate between normal and malicious traffic more effectively.

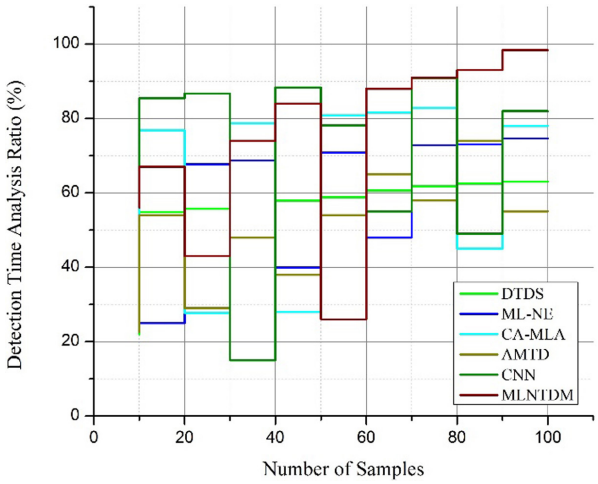

Fig. 7. Detection time analysis

In Figure 7, the responsiveness and performance of the MLNTDM in identifying and mitigating cyber threats in IoT networks are evaluated using detection time and a search. This observation measures the time it takes for the MLNTDM to become aware of and react to uncommon network interest, starting from when the suspicious interest commences until an alert or countermeasure is issued. The detection time observed sheds light on the version's chance detection competencies by performing systematic experiments and simulations underneath one-of-a-kind community conditions and assault situations, producing 97.5%. When measuring the MLNTDM's detection time performance, different factors are considered, together with the complexity of the assault, the volume of community site companies, and the computational sources available for analysis. Contrast evaluation towards existing detection structures and benchmarks helps test the MLNTDM's responsiveness and discover optimization areas. Sensitivity analysis is achieved to test how agreeably the MLNTDM's detection time holds up beneath various setups and settings. Optimizing the algorithmic layout and deployment approach of the MLNTDM to minimize detection latency and enhance the general security posture of IoT networks is heavily dependent on the findings from the detection time examination. Cybersecurity responses designed to cope with the precise threats discovered in IoT environments may be stepped forward with the help of complete detection time analyses. The reduced detection time is due to the efficient pre-processing and feature extraction processes, combined with the optimized ML algorithms that allow for rapid analysis and threat identification.

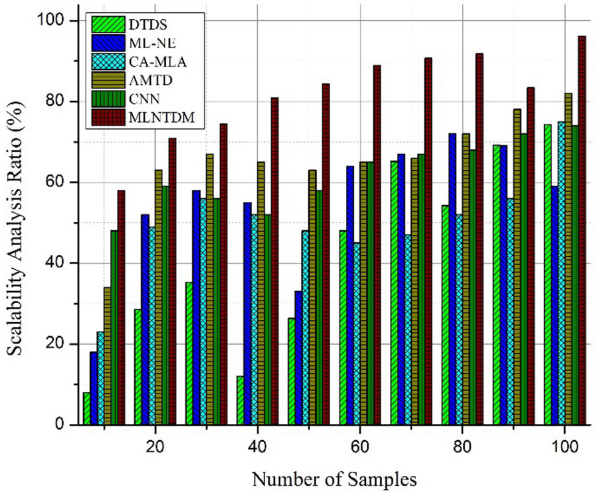

Fig. 8. Scalability analysis

In Figure 8, scalability analysis is vital to investigating the efficacy and effectiveness of an MLNTDM's evolution for IoT networks. Here, scalability is how the model can maintain working properly because it did earlier than, even if the range of nodes and the number of records transferred across them grow. The discern covers several grounds, such as how well the version handles computationally heavy incoming visitor's streams, how properly it handles greater datasets without sacrificing detection accuracy, and how well it adjusts to modifications in network layout and traffic patterns, producing 91.4%. The model's memory and processing strength desires are evaluated as part of the scalability evaluation, which aims to spot any bottlenecks that could reduce the model's performance when run at large scales. Researchers and practitioners can optimize the ML configuration and deployment method via thorough scalability evaluation to improve operational resilience and community safety in IoT environments. This will allow for green and dependable network site companies' detection. The scalability is achieved through the model's distributed processing architecture and its ability to efficiently handle large volumes of MQTT traffic without significant performance degradation.

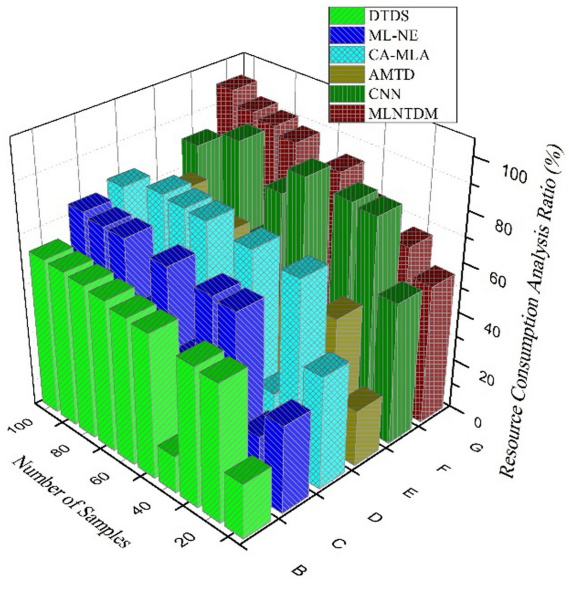

Fig. 9. Resource consumption analysis

In Figure 9, an MLNTDM advanced, especially for IoT networks, should undergo aid intake analysis to evaluate efficiency and applicability. During its operation, the version's use of different computational sources, including CPU, memory, and storage, is evaluated in this analysis. Regarding IoT installations, wherein assets are often restrained, knowing how sources are fed is crucial for improving the version's efficiency and reducing infrastructure prices. Optimization possibilities, consisting of strategies to compress models, hardware acceleration, or algorithmic enhancements, might be found by measuring resource use below diverse situations that produce 96.8%. To ensure that the infrastructure that backs up the detection version can manage operational demands without losing sources, aid consumption evaluation is a device to have the disposal. In addition, the analysis sheds light on the version's scalability, showing that it can control larger datasets and growing workloads without running out of ability. Stakeholders could make better choices concerning the design, deployment, and optimization of responses for detecting community companies in IoT environments using device mastering via acting thorough consumption analyses. This will cause better community security and operational performance. The lower resource consumption is due to the lightweight nature of the model's design, which includes efficient algorithms and minimalistic data processing techniques tailored for MQTT traffic.

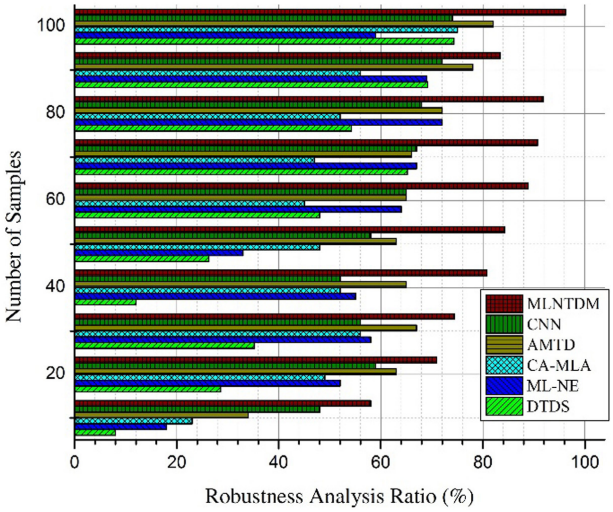

Fig. 10. Robustness analysis

In Figure 10, robustness analysis is of utmost importance in examining the efficacy and dependability of an MLNTDM designed for IoT networks. It involves trying out the model's robustness and capability to preserve detection accurately in the face of modifications in the network, as opposed to attacks. IoT community safety is especially prone to hostile assaults, which can misclassify information or regulate community site companies to avoid detection. As part of a robustness analysis, the version is examined under diverse assault scenarios to discover weaknesses and create responses. In addition, to avoid fake positives and negatives, the version must be resilient to noise and abnormalities commonplace in IoT contexts. Such issues include voice interference or sensor problems. The other aspect tested under robustness analysis was how well it adapts to changing networking conditions by introducing new company policies or adding more IoT devices, leading up to 97.3%. In addition, stakeholders can confirm its dependability by conducting extensive testing and validation techniques, ensuring effectiveness in detecting and preventing

vulnerability cases in real IoT deployments. Moreover, robustness analysis can be used for fine-tuning or optimizing a detection model, thus making it even more resilient and effective against fresh cybercrime threats. Robustness analysis is important in improving the security and reliability of IoT networks by ensuring that positioning service detection based on ML is effective and resilient. The robustness is due to the comprehensive training dataset and the model's ability to learn and adapt to new attack patterns over time, enhancing its detection capabilities.

Generally, the MLNTDM is reliable and efficient when protecting IoT systems from cyber attacks, as shown during a comprehensive evaluation across several dimensions. This model, therefore, employs modern ML technologies that recognize and halt malicious activities, providing improved safety for the IoT environment.

5 CONCLUSION

Ultimately, the recommended MLNTDM, a progressed system mastering-based totally Community Company's detection version, is a massive step forward in assembling the vital demand for robust cybersecurity in IoT networks. This model is applicable in detecting and mitigating ability cyber-attacks in opposition to IoT devices by focusing on the MQTT protocol, which is fundamental to the IoT environment. Assuring scalability and performance in hazard identity, the model demonstrates its appropriateness for useful resource-limited IoT packages using low-power consumption ML strategies. The effectiveness of the MLNTDM in detecting and countering attacks on the IoT software layer is tested through sizable testing and assessment, underscoring its promise for enhancing community security. The version does an exquisite job of detecting malicious traffic and stopping network attacks by amassing and analyzing information on community flows. Experiment findings show advanced accuracy in identifying and decreasing cyber threats in MQTT-based totally IoT networks, confirming the superiority of the notified method. The experimental results show that the suggested MLNTDM model increases the accuracy analysis of 98.7%, detection time analysis of 97.5%, scalability analysis of 91.4%, resource consumption analysis of 96.8%, and robustness analysis of 97.3% compared to other existing models. The MLNTDM shows ability as a destiny defense mechanism in opposition to complicated cyber threats, intending to help ensure the safe and dependable use of IoT generation throughout many industries. The MLNTDM can be adapted for various IoT environments beyond smart homes, including industrial IoT (IIoT), healthcare IoT, and smart cities. Each of these sectors has unique security challenges that can benefit from the model's robust detection and mitigation capabilities.

6 REFERENCES

[1] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 9395–9409, 2022. https://doi.org/10.1016/j.aej.2022.02.063

[2] M. Arun, D. Barik, and S. S. Chandran "Exploration of material recovery framework from waste – A revolutionary move towards clean environment," *Chemical Engineering Journal Advances*, vol. 18, p. 100589, 2024. https://doi.org/10.1016/j.ceja.2024.100589

[3] S. Sriram, R. A. V. I. Vinayakumar, M. Alazab, and K. P. Soman, "Network flow based IoT botnet attack detection using deep learning," in *IEEE INFOCOM 2020 – IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 189–194. https://doi.org/10.1109/INFOCOMWKSHPS50562.2020.9162668

[4] R. H. Hwang, M. C. Peng, C. W. Huang, P. C. Lin, and V. L. Nguyen, "A unsupervised deep learning model for early network traffic anomaly detection," *IEEE Access*, vol. 8, pp. 30387–30399, 2020. https://doi.org/10.1109/ACCESS.2020.2973023

[5] M. Ahmad, Q. Riaz, M. Zeeshan, H. Tahir, S. A. Haider, and M. S. Khan, "Intrusion detection in internet of things using supervised machine learning based on application and transport layer features using UNSW-NB15 data-set," *Journal on Wireless Communications and Networking*, vol. 2021, 2021. https://doi.org/10.1186/s13638-021-01893-8

[6] L. Nie *et al.*, "A reinforcement learning-based network traffic prediction mechanism in intelligent Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2169–2180, 2020. https://doi.org/10.1109/TII.2020.3004232

[7] M. A. Rahman, A. T. Asyhari, L. S. Leong, G. B. Satrya, M. H. Tao, and M. F. Zolkipli, "Scalable machine learning-based intrusion detection system for IoT-enabled smart cities," *Sustainable Cities and Society*, vol. 61, p. 102324, 2020. https://doi.org/10.1016/j.scs.2020.102324

[8] N. Islam *et al.*, "Towards machine learning based intrusion detection in IoT networks," *Computers, Materials & Continua*, vol. 69, no. 2, pp. 1801–1821, 2021. https://doi.org/10.32604/cmc.2021.018466

[9] L. Liu, P. Wang, J. Lin, and L. Liu, "Intrusion detection of imbalanced network traffic based on machine learning and deep learning," *IEEE Access*, vol. 9, pp. 7550–7563, 2020. https://doi.org/10.1109/ACCESS.2020.3048198

[10] S. Dong, Y. Xia, and T. Peng, "Network abnormal traffic detection model based on semi-supervised deep reinforcement learning," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4197–4212, 2021. https://doi.org/10.1109/TNSM.2021.3120804

[11] Q. Abu Al-Haija and A. Al-Badawi, "Attack-aware IoT network traffic routing leveraging ensemble learning," *Sensors*, vol. 22, no. 1, p. 241, 2021. https://doi.org/10.3390/s22010241

[12] M. S. Mahmood and A. B. Al Dabagh, "Improving IoT security using lightweight based deep learning protection model," *Tikrit Journal of Engineering Sciences*, vol. 30, no. 1, pp. 119–129, 2023. https://doi.org/10.25130/tjes.30.1.12

[13] J. Lam and R. Abbas, "Machine learning based anomaly detection for 5G networks," *arXiv preprint arXiv:2003.03474*, 2020.

[14] G. Abdelmoumin, D. B. Rawat, and A. Rahman, "On the performance of machine learning models for anomaly-based intelligent intrusion detection systems for the Internet of Things," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4280–4290, 2021. https://doi.org/10.1109/JIOT.2021.3103829

[15] D. K. Reddy, H. S. Behera, J. Nayak, P. Vijayakumar, B. Naik, and P. K. Singh, "Deep neural network-based anomaly detection in Internet of Things network traffic tracking for the applications of future smart cities," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 7, 2021. https://doi.org/10.1002/ett.4121

[16] O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, "A review on machine learning–based approaches for Internet traffic classification," *Annals of Telecommunications*, vol. 75, pp. 673–710, 2020. https://doi.org/10.1007/s12243-020-00770-7

[17] A. K. Sahu, S. Sharma, M. Tanveer, and R. Raja, "Internet of Things attack detection using hybrid Deep Learning Model," *Computer Communications*, vol. 176, pp. 146–154, 2021. https://doi.org/10.1016/j.comcom.2021.05.024

[18] Q. Abu Al-Haija, M. Krichen, and W. Abu Elhaija, "Machine-learning-based darknet traffic detection system for IoT applications," Electronics, vol. 11, no. 4, p. 556, 2022. https://doi.org/10.3390/electronics11040556

[19] Salman, Ola, Imad H. Elhajj, Ali Chehab, and Ayman Kayssi, "A machine learning based framework for IoT device identification and abnormal traffic detection," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 3, 2022. https://doi.org/10.1002/ett.3743

[20] R. Kumar, M. Swarnkar, G. Singal, and N. Kumar, "IoT network traffic classification using machine learning algorithms: An experimental analysis," *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 989–1008, 2021. https://doi.org/10.1109/JIOT.2021.3121517

[21] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: A malicious Bot-IoT traffic detection method in IoT network using machine-learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2020. https://doi.org/10.1109/JIOT.2020.3002255

[22] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, "Anomaly-based intrusion detection system for IoT networks through deep learning model," *Computers and Electrical Engineering*, vol. 99, p. 107810, 2022. https://doi.org/10.1016/j.compeleceng.2022.107810

[23] R. Gattu, "Network traffic dataset," Kaggle, 2023. https://www.kaggle.com/datasets/ravikumargattu/network-traffic-dataset

[24] R. Buenrostro-Mariscal, P. C. Santana-Mancilla, O. A. Montesinos-López, J. I. Nieto Hipolito, and L. E. Anido-Rifon, "A review of deep learning applications for the next generation of cognitive networks," *Applied Sciences*, vol. 12, no. 12, p. 6262, 2022. https://doi.org/10.3390/app12126262

7 AUTHORS

Mazen Alzyoud is with the Computer Science Department, Al-Bayt University, Mafraq, Jordan (E-mail: malzyoud@aabu.edu.jo).

Najah Al-shanableh is with the Computer Science Department, Al-Bayt University, Mafraq, Jordan (E-mail: najah2746@aabu.edu.jo).

Eman Nashnush is with the Computer Science Department, University of Salford, United Kingdom (E-mail E.Nashnush1@edu.salford.ac.uk).

Rabah Shboul is with the Computer Science Department, Al-Bayt University, Mafraq, Jordan (E-mail: rabahshboul@aabu.edu.jo).

Raed Alazaidah is with the Computer Science Department, Zarqa University, Zarqa, Jordan (E-mail: razaidah@zu.edu.jo).

Ghassan Samara is with the Computer Science Department, Zarqa University, Zarqa, Jordan (E-mail: gsamara@zu.edu.jo).

Safaa Alhusban is with the Computer Science Department, Al-Bayt University, Mafraq, Jordan (E-mail: Safaa.husban@gmail.com).