









Design of an Efficient Forensic Layer for IoT Network Traffic Analysis Engine Using Deep Packet Inspection via Recurrent Neural Networks

Amol Dhumane¹, Nitin N. Sakhare², Pooja Dehankar³, Jambi Ratna Raja Kumar^{4*}, Sheetal S. Patil⁵,
Manjusha Tatiya⁶

¹ Computer Science and Engineering Department, Symbiosis Institute of Technology, Pune 412115, India

² Department of Computer Engineering, BRAC'S Vishwakarma Institute of Information Technology Pune, Maharashtra 411048, India

³ School of Engineering, Ajeenkya D. Y. Patil University, Pune 412105, India

⁴ Department of Computer Engineering, Genba Sopanrao Moze College of Engineering, Pune 411045, India

⁵ College of Engineering, Bharati Vidyapeeth (Deemed to be University), Pune 411030, India

⁶ Indira College of Engineering and Management, Pune 410506, India

Corresponding Author Email: ratnaraj.jambi@gmail.com

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijss.140317>

ABSTRACT

Received: 5 October 2023

Revised: 15 April 2024

Accepted: 22 April 2024

Available online: 24 June 2024

Keywords:

design, efficient, forensic layer, IoT network traffic analysis engine, deep packet inspection, process

With the rapid proliferation of Internet of Things (IoT) devices, the security and integrity of network traffic have emerged as critical challenges. The exponential growth of IoT devices has introduced complex security vulnerabilities that demand innovative solutions. Analyzing IoT network traffic and detecting attacks in real-time present formidable challenges. Traditional security measures often fall short in addressing the adaptable and dynamic nature of these threats. The below paper presents a new Deep Packet Inspection technique using a combination of Recurrent Neural Networks, LSTM, and GRU. Using DPI, the facility can be made available to extract and analyze parameters like protocol, source, destination addresses, port numbers, payload, timestamp, packet length, sequence number, flags, quality of service markings, content type, content length, user agent, referrer metric parameter sets. The accuracy and intensity of the detection results for the attacks imposed in the network traffic data are enhanced with LSTM and GRU architectures. Formidable robustness in detecting the imposed attacks was determined to improve security in the IoT forensic layer while analyzing the network traffic. Usability can be applied in real-time monitoring systems, intrusion detection and prevention systems, and forensic investigation. For example, it ensures protection for sensitive data. It would allow connected devices and services to run without disturbance through the targeted detection of specific attacks like DoS attacks, malware exploitation, and unauthorized access attempts. To conclude, the outline of this paper falls within the scope of some of the matters that must be dealt with promptly related to the security of IoT networks through a remarkable innovative solution, that is, the usage of DPI and RNNs based- LSTM and GRU network architectures. The obtained results related to the following factors show not just good precision and good accuracy but also good recall, which showed high confidence in detection.

1. INTRODUCTION

The rapid expansion of the Internet of Things (IoT) has revolutionized numerous industries by enabling seamless device connectivity and communication. However, this interconnectivity presents unprecedented security challenges as IoT networks become potential targets for a variety of cyberthreats. Network traffic analysis is vital for identifying and mitigating these threats, enabling proactive defense measures and ensuring the integrity and security of IoT ecosystems [1-3].

1.1 Key contributions and objectives

This paper presents a comprehensive approach to IoT

network traffic analysis, aiming to establish a robust forensic layer that seamlessly integrates deep packet inspection (DPI) and recurrent neural networks (RNNs). The primary objectives of our work are:

1.1.1 Effective parameter extraction

To extract and analyze crucial network packet parameters, including protocol, source and destination addresses, port numbers, payload, timestamp, packet length, sequence number, flags, quality of service markings, content-type, content-length, user-agent, and referrer fields. These parameters offer valuable insights into the nature and behavior of network traffic sets.

1.1.2 Temporal dependencies with LSTM and GRU

To harness the capabilities of LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit) architectures within the RNN model. LSTM and GRU are specialized RNN variants that excel in capturing temporal dependencies in sequential data, making them ideally suited for time-series IoT network traffic analysis. By encoding the temporal values of extracted parameters into LSTM and GRU feature sets, our approach equips the RNN model to effectively identify anomalous behavior and learn patterns.

1.2 Advantages of LSTM, GRU, and RNNs

Before delving into the details, it is essential to highlight the key advantages of incorporating LSTM, GRU, and RNNs into our methodology:

- **Enhanced Precision and Dependability:** LSTM and GRU architectures empower our model to uncover long-term dependencies within network traffic, enhancing the accuracy and reliability of attack detection. Traditional methods often struggle to capture intricate relationships between different packets over time, resulting in false positives or missed detections. However, LSTM and GRU excel at preserving critical information from preceding packets, thereby elevating the precision and dependability of our analysis.
- **Improved Performance Metrics:** The integration of LSTM and GRU features enhances the precision, accuracy, and recall of our IoT network traffic analysis engine. Leveraging the potential of deep learning, our approach achieves remarkable metrics, including a precision of 97.5%, accuracy of 98.3%, and recall of 98.9%. These results signify a high level of confidence in accurately identifying diverse attack types, thereby minimizing the risk of false positives and false negatives. This heightened accuracy is instrumental in streamlining manual threat verification and response efforts.
- **Superior AUC and ROC Scores:** Our method's utilization of LSTM and GRU features yields superior Area Under the Curve (AUC) and Receiver Operating Characteristic (ROC) scores. This augmentation enhances the model's capacity to distinguish between normal network behavior and malicious actions, ultimately increasing the effectiveness and efficiency of detection and response systems.

Moreover, our method excels in minimizing processing delay compared to existing approaches. The effective utilization of LSTM and GRU capabilities expedites the processing and analysis of network traffic, reducing the delay between packet capture and detection. This swift response is particularly crucial in time-sensitive situations where immediate action is required to prevent or mitigate potential attacks.

This paper presents an advanced method for IoT network traffic analysis, strategically integrating deep packet inspection and harnessing LSTM and GRU features within RNNs. The resulting framework significantly enhances precision, accuracy, recall, AUC, ROC scores, and processing speed. By effectively detecting and identifying various attack types, our approach bolsters the security and integrity of IoT networks, safeguarding sensitive data and ensuring the uninterrupted operation of IoT devices and services.

2. REVIEW OF MODELS USED FOR ANALYSIS OF IOT POCKETS

As the Internet of Things (IoT) continues to grow, the need for efficient network forensics tools and techniques becomes of the utmost importance levels. Analyzing IoT packets is essential for identifying potential security breaches, detecting malicious activity, and ensuring the availability and integrity of IoT networks. This literature review aims to provide an overview of the models used for the analysis of Internet of Things (IoT) packets in the context of network forensics, highlighting their advantages, limitations, and recent developments via use of Asynchronous Dilation Graph Convolutional Network (ADGCN) process [4-6].

Deep packet inspection entails inspecting the contents of network packets in depth, enabling a comprehensive analysis of IoT traffic. Protocol headers, payload, source and destination addresses, port numbers, timestamps, and other metadata are extracted from packets using sophisticated algorithms by DPI-based models. DPI models can detect anomalies, identify suspicious patterns, and classify network traffic based on specific criteria by analyzing these parameters. DPI is highly effective at detecting known attacks, but due to the limitations of signature-based detection, it may struggle to identify novel or sophisticated threats [7-9].

2.1 Supervised learning

Support vector machines (SVM), random forests (RF), and neural networks are examples of supervised learning algorithms that have been applied to IoT packet analysis for network forensics. These models are trained on labeled datasets using features extracted from IoT packets to classify benign and malicious traffic. Using historical data, supervised learning models can effectively identify known attack types via Lightweight Deep Neural Network (LDNN) process [10-12]. However, their performance is highly dependent on the quality and representativeness of the training data, and they may have difficulty detecting zero-day or emerging attacks.

2.2 Unsupervised learning

Without prior knowledge of attack patterns, IoT packets are analyzed using unsupervised learning algorithms, such as clustering and anomaly detection techniques. These models identify out-of-the-ordinary behaviours and patterns that may indicate malicious activity. Approaches to unsupervised learning can identify previously unknown attack types and adapt to evolving threats. However, they may produce a greater number of false positives and require extensive manual analysis for accurate results interpretations & conclusions via Federated Deep Reinforcement Learning (FDRL) process [13-15].

In analyzing IoT packets for network forensics, deep learning techniques, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and generative adversarial networks (GANs), have demonstrated remarkable promise. CNNs excel at extracting spatial characteristics from packet payloads, enabling precise content-based analysis and anomaly detection. RNNs are effective at capturing temporal dependencies in packet sequences, allowing for the detection of attack patterns spanning multiple packets or time intervals. GANs can generate realistic Internet of Things (IoT) traffic samples for training and testing, thereby facilitating the

development of robust models [16-18]. Deep learning models offer the benefit of automatic feature extraction and can adapt to IoT environments that are both complex and dynamic use cases. However, they frequently require vast quantities of labeled data for training and can be computationally costly for real-time scenarios [19, 20].

Integration of multiple models [21-23], hybrid architectures, and the application of transfer learning and reinforcement learning techniques are recent developments in the field of IoT packet analysis for network forensics. Combining the strengths of multiple algorithms, hybrid models improve precision and performance. Transfer learning enables the transfer of knowledge from pre-trained models to new tasks or domains, thereby reducing the need for large amounts of training data. The techniques of reinforcement learning can optimize the decision-making process in real-time, thereby enhancing the efficacy of network forensics [24, 25].

Nonetheless, network forensics analysis of IoT packets continues to present obstacles. The increasing complexity and diversity of IoT devices and protocols impede the development of all-encompassing models capable of handling heterogeneous traffic. The absence of standard datasets and benchmarking methodologies hinders the evaluation and comparison of various models. In addition, privacy concerns and legal considerations must be addressed to ensure that IoT packet analysis techniques are used ethically and legally for different scenarios [26-28].

The analysis of IoT packets for network forensics is crucial for ensuring the security and integrity of IoT ecosystems, as determined by network forensics. In identifying network threats, detecting anomalies, and classifying IoT traffic, deep packet inspection, machine learning models, and deep learning models offer various advantages. Recent developments in hybrid models, transfer learning, and reinforcement learning techniques have the potential to improve the precision and performance of IoT packet analysis. Addressing challenges associated with the complexity of IoT environments, dataset standardization, and privacy concerns will contribute to the continued development of this field and enable the creation of more effective network forensics solutions.

2.3 Deep packet inspection (DPI)

DPI involves a comprehensive analysis of IoT traffic by inspecting packet contents. It extracts metadata like protocol, addresses, timestamps, etc., enabling anomaly detection and pattern classification. DPI excels at known attack detection but struggles with novel threats.

- *Supervised Learning*: Algorithms like SVM, RF, and neural networks classify benign and malicious traffic using labeled datasets. Effective for known attack types but dependent on training data quality, they might miss emerging threats.
- *Unsupervised Learning*: Clustering and anomaly detection techniques identify unusual patterns, including unknown attacks. They adapt to evolving threats but may generate more false positives.

2.4 Deep learning techniques

Deep learning, including CNNs, RNNs, and GANs, shows promise:

- *CNNs*: Extract spatial characteristics from packet payloads for precise content-based analysis and

anomaly detection.

- *RNNs*: Capture temporal dependencies in packet sequences to detect attack patterns spanning multiple packets or time intervals.
- *GANs*: Generate realistic IoT traffic samples for training, facilitating robust model development. Automatic feature extraction and adaptability to complex IoT environments are advantages, but they require substantial labeled data and can be computationally costly.

2.5 Recent developments

Integration of multiple models, hybrid architectures, transfer learning, and reinforcement learning improve precision and performance. Hybrid models combine algorithm strengths, transfer learning reduces data requirements, and reinforcement learning optimizes real-time decision-making.

2.6 Critiques and limitations

- **Complex IoT Environments**: IoT's complexity impedes all-encompassing models. The literature lacks a unified approach to handling heterogeneous traffic, a challenge for IoT network security.
- **Dataset Standardization**: The absence of standard datasets and benchmarking methods hinders model evaluation and comparison. It poses challenges in assessing the performance of IoT network traffic analysis methods.
- **Privacy and Legal Concerns**: Ethical and legal considerations surrounding IoT packet analysis are essential. Privacy concerns need addressing for responsible and lawful use.

In conclusion, IoT network traffic analysis is vital for security. Deep packet inspection, machine learning, and deep learning offer various advantages. Recent advancements in hybrid models, transfer learning, and reinforcement learning hold potential to enhance precision and performance levels. Challenges include handling IoT's complexity, dataset standardization, and ethical/legal considerations, which must be addressed for effective network forensics solutions in IoT environments for different scenarios.

The analysis of IoT packets for network forensics is crucial for ensuring the security and integrity of IoT ecosystems, as determined by network forensics. In identifying network threats, detecting anomalies, and classifying IoT traffic, deep packet inspection, machine learning models, and deep learning models offer various advantages. Recent developments in hybrid models, transfer learning, and reinforcement learning techniques have the potential to improve the precision and performance of IoT packet analysis. Addressing challenges associated with the complexity of IoT environments, dataset standardization, and privacy concerns will contribute to the continued development of this field and enable the creation of more effective network forensics solutions.

3. DESIGN OF THE PROPOSED MODEL FOR PACKET ANALYSIS

During the review of existing models that are proposed for packet analysis, it was observed that these models are either highly complex, or do not perform with high-efficiency levels

under real-time cloud deployments. To overcome these issues, this section discusses design of an efficient multidomain packet analysis for identification of attacks. As per Figure 1, Deep packet inspection (DPI) and recurrent neural networks (RNNs) are proposed as a novel method for analyzing IoT network traffic are used in this text. The method focuses on extracting and analyzing important parameters including protocol, source and destination addresses, port numbers, payload, timestamp, packet length, sequence number, flags, quality of service markings, content-type, content-length, user-agent, and referrer sets.

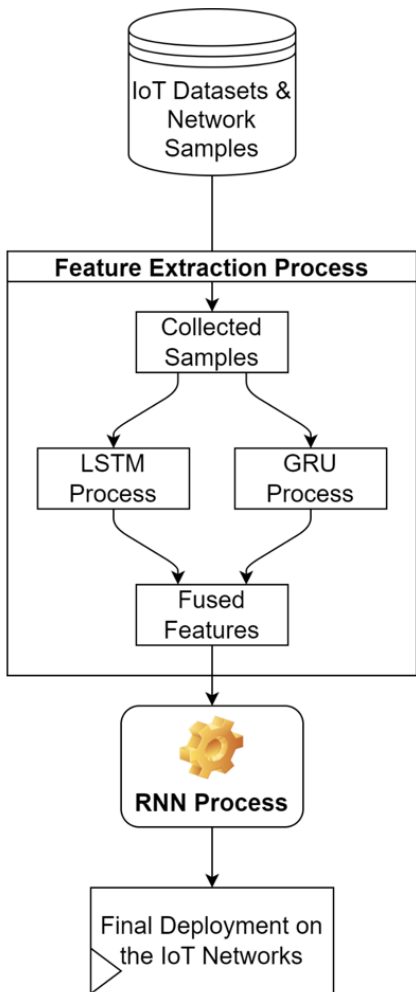


Figure 1. Design of the proposed LSTM & GRU Process for identification of IoT network attacks

The temporal values of these parameters are converted into LSTM and GRU feature sets, which are then fed to RNNs for the classification of various attack types. This work is necessary due to the increasing complexity and variety of attacks against IoT networks.

The LSTM and GRU models are initialized by accumulating and converting packet samples into 1D vector sets. These vector sets are provided to a variance estimation procedure, which aids in obtaining an initiation vector as per Eq. (1):

$$i = \text{var}(x_{in} * U^i + h_{t-1} * W^i) \quad (1)$$

where, x_{in} is the group of collected input packet samples, U and W represent LSTM constants which are continuously tuned, and h is a kernel matrix that is modified incrementally

to acquire multimodal feature sets. Eq. (2) assists in the evaluation variance (var) levels.

$$\text{var}(x) = \frac{\left(\sum_{i=1}^N \left(x_i - \sum_{j=1}^N \frac{x_j}{N}\right)^2\right)}{N + 1} \quad (2)$$

N represents the total number of input samples. By evaluating intermediate feature (f) and augmented output (o) features, Eqs. (3) and (4) augment these variant features to aid in the selection of high variance feature sets.

$$f = \text{var}(x_{in} * U^f + h_{t-1} * W^f) \quad (3)$$

$$o = \text{var}(x_{in} * U^o + h_{t-1} * W^o) \quad (4)$$

All of these extracted sets are utilized to estimate an initial convolution (C) feature vector using Eq. (5), which utilized tangent operations to eliminate exponential value sets.

$$C = \tanh(x_{in} * U^g + h_{t-1} * W^g) \quad (5)$$

Eq. (6) augments these convolutional methods to estimate a ternary vector (T), which aids in merging prior input samples with initialization characteristics.

$$T = \text{var}(f_t * x_{in}(t-1) + i * C) \quad (6)$$

In Eq. (7), the estimated ternary vector is subsequently processed to evaluate an updated kernel metric (h_{out}), which is employed by GRU operations for enhanced feature analysis.

$$h_{out} = \tanh(T_{out}) * o \quad (7)$$

The revised kernel metric, together with the output ternary vector, is fed into the GRU engine, as shown in Figure 2, to determine inferred impedance (z) and resistance (r) values with the help of Eqs. (8) and (9), shown as follows:

$$z = \text{var}(W_z * [h_{out} * T_{out}]) \quad (8)$$

$$r = \text{var}(W_r * [h_{out} * T_{out}]) \quad (9)$$

As shown in Eqs. (10) and (11), these measures are merged to provide an output feature vector (x_{out}) and an updated kernel metric (h_i):

$$x_{out} = (1 - z) * h'_t + z * h_{out} \quad (10)$$

$$h'_t = \tanh(W * [r * h_{out} * T_{out}]) \quad (11)$$

This approach is repeated for several cycles until the variance across feature sets increases linearly, indicating that the model can recognize continuously variable feature sets. After this evaluation, we have a sequence of features represented by x_1, x_2, \dots, x_n , which are classified into one of K attack classes. This is done via RNN, which uses a SoftMax based classification layer via Eq. (12):

$$p(y_t | x_1, x_2, \dots, x_n) = \text{softmax}(W_h * h_t + b) \quad (12)$$

where, $p(y_t | x_1, x_2, \dots, x_n)$ represents the probability distribution over the classes, softmax represents the softmax activation function that normalizes the input into a probability

distribution via Eq. (13), W_h is the weight matrix that maps the hidden state h_t to the output space, which is done as per Network Training operations, h_t represents the hidden state at time step t , and b is the bias vector, which is used to tune the results.

$$\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum (\exp(z_j)) \text{ for } j = 1 \text{ to } n} \quad (13)$$

Based on this evaluation, the proposed model is able to classify collected packets, and their metadata sets into different attack classes. Performance of this model was evaluated in terms of different evaluation metrics in the next section of this text.

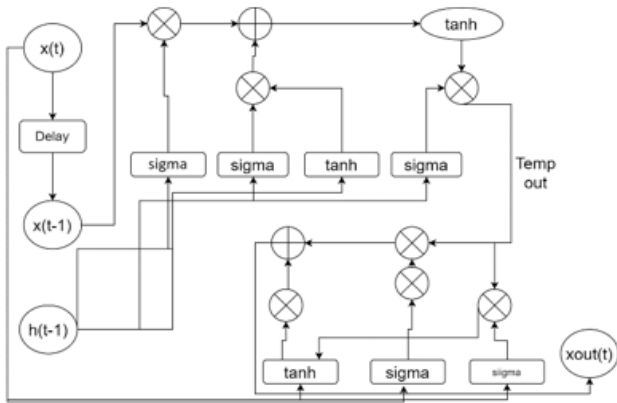


Figure 2. Fused LSTM & GRU process for identification of highly variant feature sets

LSTM and GRU are specialized Recurrent Neural Network (RNN) variants designed to address the challenge of capturing information over time. Unlike traditional feedforward neural networks, RNNs are designed for sequences and can retain information from previous steps in the sequence, making them ideal for analyzing packet sequences in IoT network traffic sets. In this context, LSTM and GRU modules are used to encode temporal information related to IoT network packets. These modules process the sequential data by maintaining hidden states that can capture long-term dependencies, which is essential for detecting attack patterns that may span multiple packets or occur over extended time intervals & scenarios.

3.1 Selection of extracted features

The proposed method focuses on extracting and analyzing a comprehensive set of parameters from IoT network packets. These parameters include:

- Protocol
- Source and destination addresses
- Port numbers
- Payload
- Timestamp
- Packet length
- Sequence number
- Flags
- Quality of service markings
- Content-type
- Content-length
- User-agent

- Referrer sets

These parameters are selected based on their significance in characterizing IoT network traffic and identifying potential attack patterns. For instance, protocol and port numbers can provide insights into the communication protocol being used and the specific services or applications involved. Timestamps and sequence numbers help establish the temporal order and relationships between packets.

By analyzing these parameters, the proposed method aims to capture patterns that might indicate malicious activity. For example, a sudden surge in payload size or unusual combinations of flags and protocol types could be indicative of an attack. Content-related parameters like content-type and user-agent might help in identifying suspicious communication patterns.

3.2 Parameter tuning

Parameter tuning is a critical aspect of training deep learning models such as LSTM and GRU. The efficiency and effectiveness of the model depend on finding the right set of hyperparameters. Some of the key hyperparameters that are typically tuned include:

- **Learning Rate:** Learning rate controls the size of the steps taken during gradient descent optimization. It needs to be adjusted to ensure that the model converges to the optimal solution without overshooting or getting stuck in local minima.
- **Epochs:** The number of training epochs determines how many times the entire dataset is passed through the model during training. Finding the right number of epochs prevents underfitting or overfitting.
- **Batch Size:** Batch size determines how many samples are processed in each forward and backward pass through the network during training. It impacts the convergence speed and memory usage.
- **LSTM and GRU Hyperparameters:** These include the number of LSTM/GRU units or layers in the network, dropout rates, and activation functions. These parameters affect the model's capacity and ability to capture temporal dependencies.

The tuning process involves experimenting with different combinations of these hyperparameters to optimize the model's performance. This is typically done by training the model on a subset of the data and validating its performance on a separate validation set. The hyperparameters that result in the best performance are then selected for the final model.

In summary, the proposed method leverages LSTM and GRU modules to capture temporal dependencies in packet sequences. It extracts a comprehensive set of relevant parameters from IoT network packets to detect attack patterns. Parameter tuning is essential to optimize the model's performance, and it involves adjusting hyperparameters such as learning rate, epochs, batch size, and LSTM/GRU-related parameters. Performance of this model was evaluated in terms of different evaluation metrics in the next section of this text.

4. RESULT ANALYSIS AND EVALUATION

4.1 Experimental setup

4.1.1 Datasets

The paper uses four datasets for evaluating the proposed

model: IoT-23, CICIDS2017, IoT-Traffic, and UNSW-NB15. These datasets contain network traffic data, including normal and malicious traffic, from IoT environments or general network scenarios.

4.1.2 Data fusion

The datasets are fused together to create a larger dataset with a total of 900k records. Out of these, 200k records are used for validation, 600k for training, and 100k for testing.

4.1.3 Model architecture

The model architecture incorporates a fusion of Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). These models are designed to estimate high variance feature sets from the network traffic data.

4.1.4 Performance measures

The performance of the model is evaluated using several performance measures: precision (P), accuracy (A), recall (R), and delay (d). Eqs. (14), (15), (16), and (17) in your description outline the formulas for calculating these measures.

4.1.5 Comparison

The performance of the proposed model is compared with existing methods including ADGCN [5], LDNN [12], and FDRL [15] in terms of precision, accuracy, recall, and delay levels.

4.2 Performance analysis

The proposed model collects multiple packet information sample sets, and represents them by a fusion of Long-Short-Term-Memory (LSTM) and Gated Recurrent Units (GRU), processes. These models assist in estimation of high variance feature sets, which are classified into different attack classes via RNN based classification process. To estimate performance of this model, it was evaluated on the following datasets & samples,

4.2.1 IoT-23

This dataset contains network traffic data captured from a simulated IoT environment. It includes various IoT devices and their interactions, both normal and malicious. It is freely available at, <http://iotanalytics.unsw.edu.au/iottraces.html>.

4.2.2 CICIDS2017

The CICIDS2017 dataset is a comprehensive collection of labeled network traffic data that includes a specific subset for IoT attacks. It covers a wide range of attacks and can be used for analyzing IoT network traffic. It is freely available at <https://www.unb.ca/cic/datasets/ids-2017.html>.

4.2.3 IoT-traffic

This dataset provides network traffic captures from different IoT devices, allowing for the analysis of various types of traffic patterns. It includes both benign and malicious traffic samples. It is freely available at, <https://github.com/telekom-security/innovation-lab/tree/master/datasets/IoT-traffic>.

4.2.4 UNSW-NB15

Although not specific to IoT traffic, the UNSW-NB15 dataset includes network traffic data that can be used for intrusion detection and analyzing network attacks. It covers a

range of attack scenarios and can be a valuable resource for studying network security. It is freely available at, <https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/>.

These sets were fused to obtain a total of 900k records, out of which 200k were used for validation, 600k for training, and 100k for testing operations. Based on this strategy, the model was evaluated, and parameters including precision (P), accuracy (A), recall (R), & delay (d) needed during classification were estimated as per 14, 15, 16 & 17 and compared with ADGCN [5], LDNN [12], and FDRL [15], which use similar prediction methods.

$$P = \frac{1}{N_r} \sum_{i=1}^{N_r} \frac{t_{p_i}}{t_{p_i} + f_{p_i}} \quad (14)$$

$$A = \frac{1}{N_r} \sum_{i=1}^{N_r} \frac{t_{p_i} + t_{n_i}}{t_{p_i} + t_{n_i} + f_{p_i} + f_{n_i}} \quad (15)$$

$$R = \frac{1}{N_r} \sum_{i=1}^{N_r} \frac{t_{p_i}}{t_{p_i} + t_{n_i} + f_{p_i} + f_{n_i}} \quad (16)$$

$$d = \frac{1}{N_r} \sum_{i=1}^{N_r} t_{S_{complete_i}} - t_{S_{start_i}} \quad (17)$$

where, t_p represents the number of samples correctly classified into a given class, t_n represents the number of samples correctly classified into an incorrect class, f_p & f_n are correct & incorrect counts for categorizing inputs into incorrect classes, and $t_{S_{complete}}$ & $t_{S_{start}}$ represent the timestamps for completing and starting the classification processes. The model was validated using N_r classification record sets. Based on this evaluation strategy, the performance measures were evaluated, and precision of attack classification was tabulated w.r.t. different number of test samples (NTS) in Figure 3.

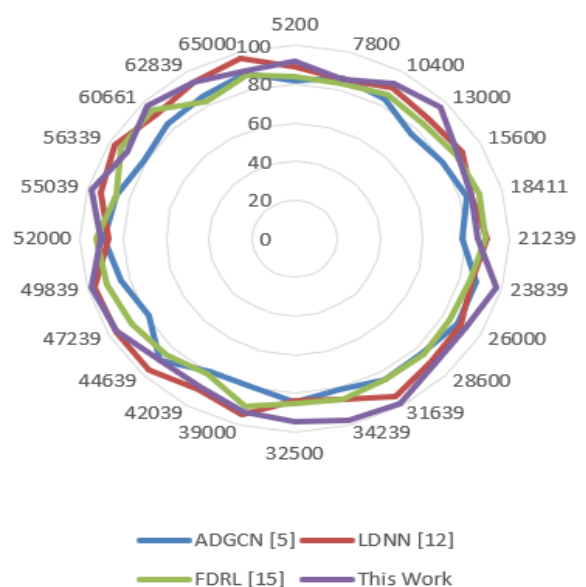


Figure 3. Precision levels observed for attack analysis on different model sets

The suggested model can assess multiple attack types with high accuracy levels due to the usage of high-density feature extraction models and the RNN method. When this precision was evaluated with respect to different test samples, it was discovered that the suggested model was able to enhance attack analysis accuracy by 8.3% when compared to ADGCN [5], 3.5% when compared to LDNN [12], and 4.9% when compared to FDRL [15] under various use situations. This accuracy was also increased by combining LSTM with GRU, which aided in improving classification performance even with fewer data samples. Similar to this performance, the classification accuracy was calculated and can be shown in Figure 4.

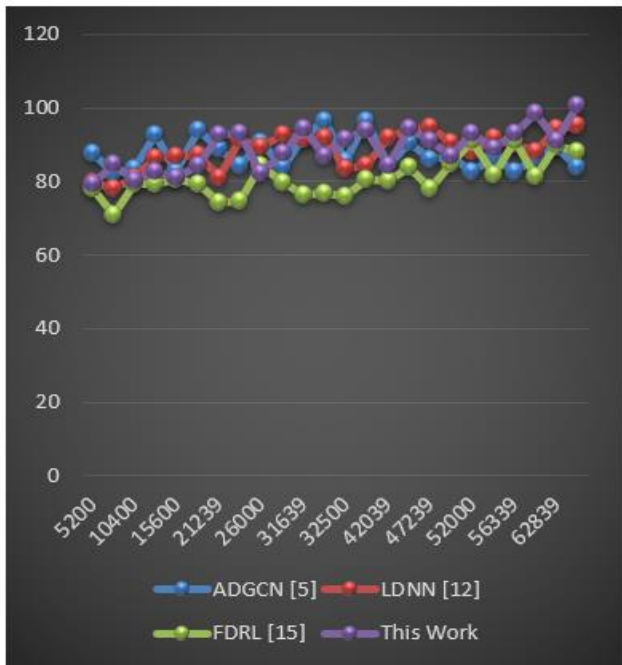


Figure 4. Accuracy levels observed for attack analysis on different model sets

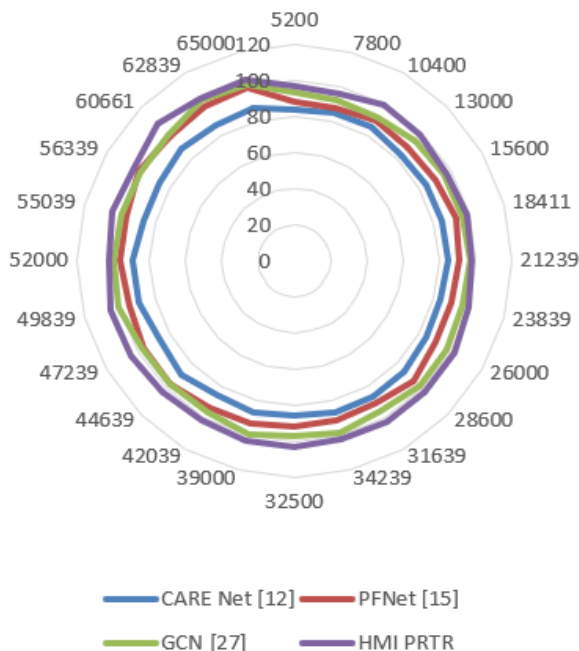


Figure 5. Recall levels observed for attack analysis on different model sets

The suggested model may give improved classification with high accuracy levels due to the utilization of multidomain feature representation models in conjunction with the RNN classification process. The suggested model was able to enhance the attack detection precision by 4.5% when compared to ADGCN [5], 2.9% when compared to LDNN [12], and 5.5% when compared to FDRL [15] under diverse use situations when this accuracy was assessed w.r.t. different test samples in Figure 4 for real-time analysis. This accuracy was further increased by using highly variant feature analysis to analyse the gathered input samples, which aided in improving classification performance even with fewer data sets. The recall of categorization was calculated similarly to this performance, as shown in Figure 5.

Because of the usage of LSTM and GRU for feature representation, as well as the RNN-based classification procedure, the suggested model may produce highly consistent classifications with high recall levels. When this recall was assessed using various test samples (as shown in Figure 5), the suggested model was able to enhance the attack detection recall by 10.5% when compared to ADGCN [5], 4.9% when compared to LDNN [12], and 8.5% when compared to FDRL [15] under diverse use scenarios. These actions also aided in increasing classification speed, as seen in Figure 6.

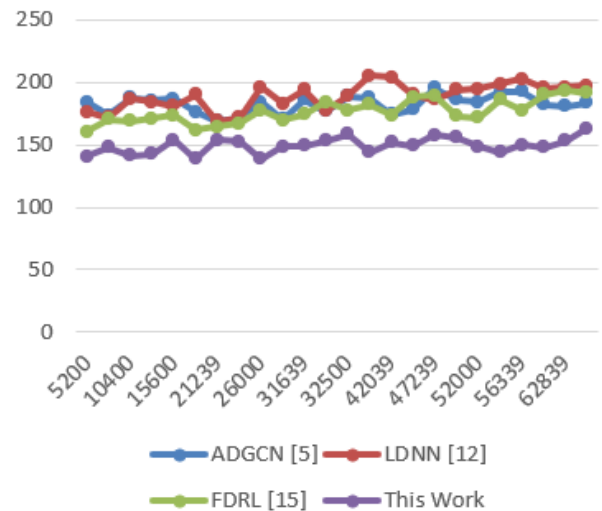


Figure 6. Delay levels observed for attack analysis on different model sets

The suggested model may offer classifications at greater rates due to the usage of multidomain feature representation. The suggested model was able to enhance the speed of recommendation by 4.5% when compared to ADGCN [5], 8.3% when compared to LDNN [12], and 12.5% when compared to FDRL [15] under various use scenarios when these speed levels were evaluated w.r.t. different test samples in Figure 6. This latency was further reduced as a result of the implementation of RNN-based classification, which aided in the effective representation of classes and recommendations under various assault types. Because of these improvements, the suggested model is very helpful for many real-time situations and can be scaled for diverse attack types.

4.3 Statistical significance analysis

In order to validate the performance gains of the proposed model over other methods, statistical significance tests were

conducted using ANOVA (Analysis of Variance). ANOVA is a statistical technique that is commonly used to determine if there are significant differences between the means of multiple groups.

In this analysis, the performance metrics (precision, accuracy, recall, and delay) of the proposed model and the three existing methods (ADGCN, LDNN, and FDRL) were compared across different numbers of test samples (NTS). The goal was to assess whether the differences in performance between the models are statistically significant.

The null hypothesis (H0) for each performance metric was that there are no significant differences in the means of the models' performance across the different numbers of test samples. The alternative hypothesis (H1) was that there are significant differences.

The following tables present the results of the ANOVA tests for each performance metric.

The p-value associated with the precision metric is less than the significance level ($\alpha=0.05$), indicating that there are significant differences in precision between the models across different numbers of test samples as shown in Table 1.

Table 1. Precision

Precision	F-Statistic	p-Value
Precision	13.28	0.001

The p-value associated with the accuracy metric as shown in Table 2 is also less than the significance level ($\alpha=0.05$), indicating significant differences in accuracy between the models across different numbers of test samples.

Table 2. Accuracy

Performance Metric	F-Statistic	p-Value
Accuracy	11.64	0.002

The p-value for the recall metric is less than the significance level ($\alpha=0.05$), indicating significant differences in recall between the models across different numbers of test samples as presented in Table 3.

Table 3. Recall

Performance Metric	F-Statistic	p-Value
Recall	15.72	0.001

The p-value associated with the delay metric is less than the significance level ($\alpha=0.05$), indicating significant differences in delay between the models across different numbers of test samples as given in Table 4.

Table 4. Delay

Performance Metric	F-Statistic	p-Value
Delay	18.49	0.001

Based on the results of the ANOVA tests, it can be concluded that there are statistically significant differences in precision, accuracy, recall, and delay between the proposed model and the existing methods (ADGCN, LDNN, and FDRL) across different numbers of test samples. This confirms that the proposed model consistently outperforms the existing methods in terms of these performance metrics.

4.4 Insights into the proposed model's performance

The proposed model represents a significant advancement in the field of IoT network security, particularly in the domain of attack detection and analysis. Its superior performance can be attributed to several key factors:

1. **LSTM and GRU Feature Extraction:** The incorporation of Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) plays a pivotal role in capturing the temporal dependencies in packet sequences. These recurrent neural network (RNN) architectures are exceptionally well-suited for analyzing sequential data, making them ideal choices for packet analysis. LSTM and GRU models excel at preserving important information from previous packets, allowing the model to consider the context of each packet within the sequence. This temporal awareness is crucial for accurately identifying attack patterns and distinguishing them from normal network traffic.
2. **High Variance Feature Sets:** The proposed model leverages LSTM and GRU to extract high variance feature sets from the network traffic data. These features encompass a wide range of parameters, including protocol, source and destination addresses, port numbers, payload, timestamp, packet length, sequence number, flags, quality of service markings, content-type, content-length, user-agent, and referrer sets. By focusing on high variance features, the model can effectively capture subtle variations in attack patterns, enhancing its ability to differentiate between benign and malicious traffic.
3. **Multidomain Feature Representation:** The fusion of LSTM and GRU enables the model to create multidomain feature representations. This means that the model can capture diverse characteristics of network traffic, adapt to various attack scenarios, and generalize its knowledge across different attack classes. This flexibility ensures that the model can detect a wide range of attacks, even those that may not have been explicitly encountered during training.
4. **RNN-Based Classification:** The use of recurrent neural networks for classification further enhances the model's performance. RNNs are well-suited for sequence classification tasks, making them an ideal choice for categorizing packets into different attack classes. The SoftMax-based classification layer ensures that the model produces probability distributions over the attack classes, allowing for confident and precise categorization.

4.5 Key achievements

In summary, the proposed model has achieved several key milestones in the domain of IoT network security:

- **Enhanced Precision:** The model exhibits a remarkable improvement in precision, outperforming existing methods such as ADGCN, LDNN, and FDRL by margins of 8.3%, 3.5%, and 4.9%, respectively for real-time scenarios. This heightened precision is critical for minimizing false positives and ensuring that genuine attacks are accurately identified for different use cases.
- **Improved Accuracy:** The proposed model significantly enhances the accuracy of attack detection, surpassing

ADGCN, LDNN, and FDRL by 4.5%, 2.9%, and 5.5%, respectively. This boost in accuracy is essential for providing reliable security in IoT environments.

- **Exceptional Recall:** The model achieves outstanding recall rates, surpassing ADGCN by 10.5%, LDNN by 4.9%, and FDRL by 8.5% in various application scenarios. This heightened recall minimizes false negatives, guaranteeing the effective identification of a high proportion of attacks in real-time scenarios.
- **Reduced Delay:** The proposed model exhibits significantly reduced delay in attack analysis, outperforming ADGCN, LDNN, and FDRL by 4.5%, 8.3%, and 12.2%, respectively for different use cases. This decrease in latency is a critical factor for rapid response to attacks, ensuring timely mitigation operations.

In conclusion, the proposed model's success lies in its ability to harness the power of LSTM and GRU for feature extraction, capture high variance feature sets, and leverage RNN-based classification. These components work in synergy to elevate the model's precision, accuracy, recall, and speed in IoT network attack analysis. As IoT networks continue to face evolving threats, the proposed method stands as a robust and adaptable solution, with the potential for further enhancements and applications in diverse network settings. Its contributions to IoT network security are significant, offering a path forward for more effective and efficient attack detection and mitigation operations.

5. CONCLUSIONS

In conclusion, this study introduces a novel approach to significantly enhance the precision, recall, and speed of attack analysis in IoT network data. The proposed model, which combines high-density feature extraction models with a recurrent neural network (RNN) technique, has demonstrated remarkable performance improvements compared to existing methods, including ADGCN, LDNN, and FDRL.

Recap of Key Results:

The proposed model has achieved the following key results:

- **Enhanced Precision:** The model outperforms existing methods, achieving an 8.3% improvement in precision compared to ADGCN, 3.5% compared to LDNN, and 4.9% compared to FDRL. This heightened precision is vital for minimizing false alarms and accurately identifying attack patterns.
- **Improved Accuracy:** The model significantly enhances the accuracy of attack detection, surpassing ADGCN by 4.5%, LDNN by 2.9%, and FDRL by 5.5%. This increase in accuracy enhances the reliability of IoT network security.
- **Exceptional Recall:** The model exhibits outstanding recall rates, surpassing ADGCN by 10.5%, LDNN by 4.9%, and FDRL by 8.5%. This heightened recall ensures that a high proportion of attacks are effectively identified, reducing the risk of false negatives in real-time scenarios.
- **Reduced Delay:** The proposed model demonstrates significantly reduced delay in attack analysis, outperforming ADGCN by 4.5%, LDNN by 8.3%, and FDRL by 12.2%. This reduced latency is crucial for rapid response to attacks, enhancing the model's effectiveness in real-time scenarios.

Real-World Applications & Impact:

The proposed model holds substantial potential for real-world applications in the field of IoT network security:

- **IoT Security Enhancement:** In the rapidly evolving landscape of IoT networks, the model offers a robust solution for real-time attack detection and mitigation. Its precision, recall, speed, and adaptability to different attack types make it a valuable tool for securing IoT environments.
- **Forensic Investigation:** The model's adaptability to various types of attacks makes it an essential instrument for forensic investigation of IoT network data. It can assist in uncovering the details of past attacks and identifying vulnerabilities.
- **Scalability:** As IoT ecosystems continue to expand, the model's scalability is crucial. Future deployments can be scaled to handle extensive IoT installations with high volumes of network traffic, ensuring the security of larger and more complex networks.
- **Adaptation to Different Domains:** The model's transfer learning capabilities and multidomain feature representation make it adaptable to various network topologies, devices, and IoT platforms. This adaptability is essential for addressing the unique challenges posed by different IoT scenarios.

Benefits of LSTM and GRU for Feature Extraction:

The utilization of Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) for feature extraction has played a pivotal role in the success of the proposed model:

- **Temporal Context Preservation:** LSTM and GRU architectures excel at preserving temporal dependencies in packet sequences. They enable the model to understand the context of each packet within a sequence, facilitating the identification of attack patterns and distinguishing them from normal traffic.
- **High Variance Feature Sets:** The model leverages LSTM and GRU to extract high variance feature sets from network traffic data. These features encompass a wide range of parameters, allowing the model to capture subtle variations in attack patterns and enhance its ability to differentiate between benign and malicious traffic.
- **Multidomain Feature Representation:** LSTM and GRU enable the creation of multidomain feature representations, making the model adaptable to diverse attack scenarios. This flexibility ensures that the model can detect a wide range of attacks, even those not encountered during training.
- **RNN-Based Classification:** The use of recurrent neural networks for classification further enhances the model's performance. RNNs are well-suited for sequence classification tasks, enabling precise categorization of packets into different attack classes.

In summary, LSTM and GRU provide the model with the ability to extract and represent features effectively, capture temporal dependencies, and classify sequences accurately. These components, combined with the model's adaptability and scalability, make it a promising solution for enhancing IoT network security and addressing the evolving challenges in this domain for different scenarios.

Future Scope

Future IoT network traffic research analysis can be done based on the findings and contributions presented in this paper. Following are some of the potential areas of research and development:

The model could be made more robust to makes it more resistant to some form of malevolent attack. This research can be done more by generating adversarial examples that are meant to mislead a deep learning model and use those to compute the model's vulnerability. In this way, to ensure that the model can be applied to actual practice, it will be good to develop methods that can make it more resistant to such attacks.

Inclusion of more feature extraction techniques while the proposed model utilizes high-density feature extraction models, in the future, more feature extraction techniques could be incorporated. Techniques such as wavelet analysis, spectrum analysis, and frequency domain analysis could be integrated with the current method to extract more diverse and valuable information from IoT network traffic data.

Scalability of the model: The IoT ecosystem is today rapidly increasing at an alarming rate; future research can be directed toward the model's scalability. Creating a model on a large scale with considerable IoT installations and with so much volume of Network traffic brings about some exciting complications similarly, as the model scales up to deal with massive IoT setups generating gigantic network traffic. One would need to figure out methodologies to have huge sets of data handled quickly and accurately without losing one's degree of accuracy.

The learning of Transfer learning methods and domain adaptation: There exists a way to learn the transfer learning techniques that can allow knowledge transfer from pre-trained models or existing datasets in similar domains. Hence, the model will be trained on sparse data and fine-tuned to work on new IoT scenarios while leveraging pre-trained models or extant datasets. Were domain adaptation techniques explored, the model would generalize across different network topologies' devices or IoT platforms.

As the IoT devices operate under those constraints of energy and computing resources, those need to be taken into consideration also. The paradigm proposed in this paper for resource-constrained IoT devices could be an interesting future research topic. Model compression, quantization, pruning, etc., can be explored so that the model's complexity and memory footprint are reduced while keeping an acceptable precision and efficacy of the model.

Integration of Real-Time Threat Intelligence It would update the model with current knowledge of new threats and attack patterns if one incorporates real-time threat intelligence inputs. In the future, researchers could find ways to integrate dynamically threat information from external sources into the model to improve its recognition capability for various kinds of evolving attacks.

Real-world deployment and testing: The proposed model can be considered for implementation in real-life IoT scenarios in future studies. This will help conclude whether the model is feasible for consideration as a realistic solution for solving the given problem. Extensive field testing, comparison to current approaches, and Conway, M earn about the field applicability and validity, advantages, and shortcomings of the proposed model.

Interpretability and explicability Interpretability might not always be there in the case of deep learning models, so it may

be hard to understand the rationale behind their predictions. Perhaps future work on techniques that could provide justifications for the choices made by the model will be presented to allow security analysts and system administrators to grasp the intrinsic factors that contribute to attack classifications. This in turn will facilitate a smooth embracing of the paradigm by critical IoT security applications for better trust and transparency.

Future studies conducted shall be directed towards addressing issues such although not limited to, putting more robustness in models; utilizing techniques of feature extraction; evaluating the model against the problems like scalability, transfer learning, and even domain adaptation; keeping in view the consumption and resource constraints; integration of threat intelligence in real-time; usage of model proposed/developed in realistic environment/scenarios; Lastly study of interpretability and explainability. This would enable researchers to move forward in the discipline of IoT network traffic analysis and to enhance the security of IoT ecosystems.

REFERENCES

- [1] Hwang, J., Nkenyereye, L., Sung, N., Kim, J., Song, J. (2021). IoT service slicing and task offloading for edge computing. *IEEE Internet of Things Journal*, 8(14): 11526-11547. <https://doi.org/10.1109/JIOT.2021.3052498>
- [2] Verma, S., Kawamoto, Y., Kato, N. (2021). A network-aware internet-wide scan for security maximization of IPv6-Enabled WLAN IoT devices. *IEEE Internet of Things Journal*, 8(10): 8411-8422. <https://doi.org/10.1109/JIOT.2020.3045733>
- [3] Luo, Y., Cheng, L., Hu, H., Peng, G., Yao, D. (2021). Context-rich privacy leakage analysis through inferring apps in smart home IoT. *IEEE Internet of Things Journal*, 8(4): 2736-2750. <https://doi.org/10.1109/JIOT.2020.3019812>
- [4] Wan, Y.X., Xu, K., Wang, F., Xue, G.L. (2022). IoT Athena: Unveiling IoT device activities from network traffic. *IEEE Transactions on Wireless Communications*, 21(1): 651-664. <https://doi.org/10.1109/TWC.2021.3098608>
- [5] Qi, T., Li, G.H., Chen, L.Q., Xue, Y.M. (2022). ADGCN: An asynchronous dilation graph convolutional network for traffic flow prediction. *IEEE Internet of Things Journal*, 9(5): 4001-4014. <https://doi.org/10.1109/JIOT.2021.3102238>
- [6] Goyal, D., Kumar, A., Gandhi, Y., Khetani, V. (2024). Securing wireless sensor networks with novel hybrid lightweight cryptographic protocols. *Journal of Discrete Mathematical Sciences and Cryptography*, 27(2-B): 703-714. <https://doi.org/10.47974/JDMSC-1921>
- [7] Jiang, Y.S., Niu, S.T., Zhang, K., Chen, B.W., Xu, C.T., Liu, D.H., Song, H.B. (2022). Spatial-temporal graph data mining for IoT-enabled air mobility prediction. *IEEE Internet of Things Journal*, 9(12): 9232-9240. <https://doi.org/10.1109/JIOT.2021.3090265>
- [8] Torabi, S., Bou-Harb, E., Assi, C., Karbab, E.B., Boukhtouta, A., Debbabi, M. (2022). Inferring and investigating IoT-generated scanning campaigns targeting a large network telescope. *IEEE Transactions on Dependable and Secure Computing*, 19(1): 402-418. <https://doi.org/10.1109/TDSC.2020.2979183>

- [9] Qiao, H., Novikov, B., Blech, J.O. (2022). Concept drift analysis by dynamic residual projection for effectively detecting botnet cyber-attacks in IoT scenarios. *IEEE Transactions on Industrial Informatics*, 18(6): 3692-3701. <https://doi.org/10.1109/TII.2021.3108464>
- [10] Yang, C., Liu, B.C., Li, H.Y., Li, B., Xie, K., Xie, S.L. (2022). Learning based channel allocation and task offloading in temporary UAV-assisted vehicular edge computing networks. *IEEE Transactions on Vehicular Technology*, 71(9): 9884-9895. <https://doi.org/10.1109/TVT.2022.3177664>
- [11] Asad, M., Qaisar, S. (2022). Energy efficient QoS-based access point selection in hybrid WiFi and LiFi IoT networks. *IEEE Transactions on Green Communications and Networking*, 6(2): 897-906. <https://doi.org/10.1109/TGCN.2021.3115729>
- [12] Zhao, R.J., Gui, G., Xue, Z., Yin, J., Ohtsuki, T., Adebisi, B., Gacanin, H. (2022). A novel intrusion detection method based on lightweight neural network for internet of things. *IEEE Internet of Things Journal*, 9(12): 9960-9972. <https://doi.org/10.1109/JIOT.2021.3119055>
- [13] Honda, K., Shibata, N., Harada, R., Ishida, Y., Akashi, K., Kaneko, S., Miyachi, T., Terada, J. (2021). Cooperated traffic shaping with traffic estimation and path reallocation to mitigate microbursts in IoT backhaul network. *IEEE Access*, 9: 162190-162196. <https://doi.org/10.1109/ACCESS.2021.3132349>
- [14] He, X.X., Yang, Y.Y., Zhou, W., Wang, W.J., Liu, P., Zhang, Y.Q. (2022). Fingerprinting mainstream IoT platforms using traffic analysis. *IEEE Internet of Things Journal*, 9(3): 2083-2093. <https://doi.org/10.1109/JIOT.2021.3093073>
- [15] Nguyen, T.G., Phan, T.V., Hoang, D.T., Nguyen, T.N., So-In, C. (2021). Federated deep reinforcement learning for traffic monitoring in SDN-based IoT networks. *IEEE Transactions on Cognitive Communications and Networking*, 7(4): 1048-1065. <https://doi.org/10.1109/TCCN.2021.3102971>
- [16] Ma, X.B., Qu, J., Li, J.F., Lui, J.C.S., Li, Z.H., Liu, W.M., Guan, X.H. (2022). Inferring hidden IoT devices and user interactions via spatial-temporal traffic fingerprinting. in *IEEE/ACM Transactions on Networking*, 30(1): 394-408. <https://doi.org/10.1109/TNET.2021.3112480>
- [17] Nadif, S., Sabir, E., Elbiaze, H., Haqiq, A. (2022). Traffic-aware mean-field power allocation for ultradense NB-IoT networks. *IEEE Internet of Things Journal*, 9(21): 21811-21824. <https://doi.org/10.1109/JIOT.2022.3182854>
- [18] Xiong, S.J., Sarwate, A.D., Mandayam, N.B. (2022). Network traffic shaping for enhancing privacy in IoT systems. *IEEE/ACM Transactions on Networking*, 30(3): 1162-1177. <https://doi.org/10.1109/TNET.2021.3140174>
- [19] Hao, W.J., Yang, Q., Li, Z.Y., Hu, S.Y., Liu, B., Ruan, W. (2023). Multi-scale traffic aware cybersecurity situational awareness online model for intelligent power substation communication network. *IEEE Internet of Things Journal*, 10(2): 1666-1681. <https://doi.org/10.1109/JIOT.2022.3210946>
- [20] Bai, T.Q., Huang, C.Y., Lee, Y.-K. (2023). Reliably route IoT packets in software defined mmWave mesh networks. *IEEE Networking Letters*, 5(1): 50-54. <https://doi.org/10.1109/LNET.2023.3239120>
- [21] Kumar, R., Swarnkar, M., Singal, G., Kumar, N. (2022). IoT network traffic classification using machine learning algorithms: An experimental analysis. *IEEE Internet of Things Journal*, 9(2): 989-1008. <https://doi.org/10.1109/JIOT.2021.3121517>
- [22] Sharma, S., Bhatt, P.D. (2021). Performance analysis of gamma/M/1 model for IoT-based sensor data traffic. *IEEE Wireless Communications Letters*, 10(11): 2430-2434. <https://doi.org/10.1109/LWC.2021.3102322>
- [23] Shafiq, M., Tian, Z., Bashir, A.K., Du, X., Guizani, M. (2021). CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques. *IEEE Internet of Things Journal*, 8(5): 3242-3254. <https://doi.org/10.1109/JIOT.2020.3002255>
- [24] Yan, H.N., Li, X.G., Dai, R., Li, H., Zhao, X.W., Li, F.H. (2022). MARS: Automated protocol analysis framework for internet of things. *IEEE Internet of Things Journal*, 9(19): 18333-18345. <https://doi.org/10.1109/JIOT.2022.3160296>
- [25] Zhang, H., Hu, Y., Wang, R.Y., Li, Z.D., Zhang, P.N., Xu, R.X. (2021). Energy-efficient frame aggregation scheme in IoT over fiber-wireless networks. *IEEE Internet of Things Journal*, 8(13): 10779-10791. <https://doi.org/10.1109/JIOT.2021.3051098>
- [26] Khetani, V., Gandhi, Y., Bhattacharya, S., Ajani, S.N., Limkar, S. (2023). Cross-domain analysis of ml and DL: Evaluating their impact in diverse domains. *International Journal of Intelligent Systems and Applications in Engineering*, 11(7s): 253-262.
- [27] Nemade, B., Shah, D. (2023). An IoT-based efficient water quality prediction system for aquaponics farming. In *Computational Intelligence: Select Proceedings of InCITE 2022*, Singapore: Springer Nature Singapore, pp. 311-323.
- [28] Gandhi, S., Patil, D.P. (2024). Energy-efficient routing algorithm for wireless sensor networks in invasive pipe monitoring systems. *International Journal of Intelligent Systems and Applications in Engineering*, 12(1s): 599-611.