DEEP LEARNING-DRIVEN AUTOMATED IOT DEVICE IDENTIFICATION USING FULL PACKET DATA

SINGAMANENI KRISHNAPRIYA
singamanenikrishnapriya@gmail.com

Sukhvinder Singh sukh.csc@pondiuni.ac.in Pondicherry University, Pondicherry, India

ABSTRACT

The rapid proliferation of Internet of Things (IoT) devices has introduced significant challenges in network security, device management, and traffic monitoring. Accurate and automated IoT device iden- tification is critical for ensuring secure communication, anomaly detection, and enforcing access control policies. Traditional identification methods, relying on static rule-based approaches or shallow learning techniques, struggle with the increasing diversity and evolving communication patterns of IoT devices. In this study, we propose a novel deep learning-driven framework that leverages full packet data analysis to achieve robust and scalable IoT device identification. The framework integrates Convolutional Neu- ral Networks (CNNs) for spatial feature extraction and Long Short-Term Memory (LSTM) networks for temporal pattern learning, enabling it to effectively capture packet header structures, payload distributions, and sequential dependencies in IoT traffic. Additionally, dropout regularization is employed to enhance generalization and mitigate overfitting, ensuring resilience across heterogeneous IoT environments. The proposed method is evaluated using benchmark IoT datasets, including UNSW IoT-23 and NB-IoT, which demonstrate superior classification accuracy, scalability, and adaptability compared to existing approaches. Experimental results highlight the effectiveness of hybrid deep learning models in IoT security, achieving high precision and low false positive rates in device identification. This research underscores the potential of full packet data-driven deep learning approaches to fortify IoT network defenses and advance next- generation automated cybersecurity solutions.

Keywords: *Deep Learning,Full Packet Data Analysis,Convolutional Neural Networks (CNN),Long Short Term Memory (LSTM),Anomaly Detection*

1 INTRODUCTION

The Internet of Things has revolutionized many industries by connecting billions of devices that com- municate seamlessly. Smart homes, healthcare, in- dustrial automation, etc. now we all have IoT devices that are integral to modern systems. Although this proliferation of IoT devices is occurring quickly, the security of these networks is facing unprecedented challenges because IoT devices are not authorized and malicious actors are taking advantage of vulnerabilities in IoT ecosystems. A crucial first step in the secure and efficient management of IoT networks is, therefore, accurate identification of IoT devices. Cur- rent device identification techniques, e.g., traditional static rule approaches or shallow learning models, are challenged by the increasing variety of IoT devices with time-varying communication patterns [1][2].

2 LITERATURE REVIEW

With recent advances in deep learning, some promising solutions are emerging to mitigate the dif- ficulties associated with IoT device identification. Unlike most conventional methods, deep learning models can learn in a automatic manner intricate pat- terns and relationships from raw data, free from the need to acquire features by hand. Of these, mod- els that are hybrids between convolutional neural net- works (CNNs) and long-short-term memory (LSTM) networks have been distinguished as having a strong performance in tasks that benefit from spatial and temporal learning features [3]. CNNs are good at detecting spatial patterns in data; LSTMs are good at modeling sequential dependencies; this works well for IoT traffic data, which has spatial and temporal aspects [4][5].

In this study, we seek to overcome the limitations of existing methods by using a deep learning-driven framework for IoT device identification that utilizes complete packet data. Unlike earlier efforts of merely statistical features or packet headers used to repre- sent network traffic, full packet data provides a more complete representation of network traffic. The pro- posed method captures HEADER and PAYLOAD in- formation to differentiate between devices with sim- ilar communication patterns [6][7]. In addition, with the help of dropout, which is a regularization technique, the generalization ability of the model is also

improved, and it can avoid overfitting, thus achieving good performance for different data sets [8].

To validate the effectiveness of the proposed approach, experiments were carried out on two bench- mark datasets: the UNSW IoT dataset [9] and the NB-IoT dataset [10]. These datasets cover various IoT devices and communication patterns, resulting in a strong evaluation framework. On the one hand, the results show that the CNN+LSTM network architec- ture paired with complete packet data and dropout be- comes a low False Positive Rate (FPR) and achieves high accuracy across various configurations. This shows the capability of leveraging deep learning tech- niques for IoT security to deal with key problems in the large-scale identification of devices for practical purposes.

Year	Publication	Methodology	Key Findings	Datasets Used	Limitations
2019	Liu et al. (2019)	Introduced a hybrid **CNN-LSTM model** for IoT device identification, combining CNN's spatial feature extraction and LSTM's temporal modeling capabilities.	Demonstrated high accuracy in identifying IoT devices by leveraging both spatial and temporal features of traffic.	**NSL-KDD**; **UNSW-NB15**	- Struggles with new or unseen devices. - Limited scalability in large IoT networks.
2020	Sharma et al. (2020)	Proposed a deep learning model that uses **full packet data** for device identification, improving accuracy by utilizing both header and payload information.	Improved accuracy by analyzing both header and payload; effective in identifying devices with complex communication patterns.	**UNSW IoT** dataset	- Computationally expensive. - May not generalize well in environments with noise.
2020	Kwon et al. (2020)	Used a **CNN-based deep learning model** to classify IoT devices based on raw packet data.	Raw packet data provided better classification accuracy than header-only data, improving device identification.	**CICIDS 2017**; **UNSW-NB15**	- Lack of robustness in adversarial or noisy network environments. - Inconsistent performance across different types of IoT devices.

Year	Publication	Methodology	Key Findings	Datasets Used	Limitations
2021	Wang et al. (2021)	Incorporated an **attention mechanism** into deep learning models for focusing on critical features in IoT traffic.	The attention mechanism allowed the model to better focus on important patterns, improving classification accuracy.	**IoT-23; CICIDS 2017**	- Attention mechanisms increase model complexity. - May require significant training data.
2022	Zhao et al. (2022)	Proposed an **end-to-end reinforcement learning-based model** that dynamically adjusts its parameters for device identification based on network conditions.	The reinforcement learning approach helped the model adapt to varying network conditions and improve robustness.	**IoT-23; CICIDS 2018**	- High training overhead. - Potential for overfitting in highly variable networks.
2023	Gupta et al. (2023)	Developed a **multimodal deep learning model**, combining network traffic and device fingerprinting for improved accuracy.	Combining multiple data sources reduced false positives and improved identification accuracy.	**UNSW IoT; CICIDS 2017**	- Struggles with devices not present in the training dataset. - Requires large-scale labeled data for training.
2024	Xu et al. (2024)	Explored **self-supervised learning** techniques for scalable de- vice identification, enabling models to adapt without labeled data.	Self-supervised learning proved effective for identifying devices with minimal labeled data, enhancing scalability.	**CICIDS 2017; UNSW IoT**	- High computational requirements. - Limited applicability in highly diverse or adversarial net- work conditions.
2024	Zhou et al. (2024)	Proposed **unsupervised learn- ing** methods for anomaly detection and device clas- sification without labeled data.	Unsupervised models successfully detected anomalies and classified devices without needing labeled traffic.	**IoT-23; CICIDS 2017**	- Performance may degrade in highly complex network environments. - Limited ability to distinguish between similar devices.

Several inferences may be made through the literature review in this research on Deep Learning-Driven Automated IoT Device Identification Using Full Packet Data. Firstly, deep learning models, especially Convolutional Neural Networks (CNN), successfully detect patterns in packet data (Liu et al., 2022), which excel at spatial feature extraction. Additionally, long-short-term memory (LSTM) net- works have been found to be helpful in modeling the temporal dependency in packet sequences for IoT de- vices, as demonstrated in IoT device identification tasks where time series data is crucial (Khan et al., 2021).

In this work, CNN and LSTM are combined, and their strengths are used the best. CNN works on spatial features, and LSTM is excellent at analyzing temporal sequences. The hybrid approach mentioned above provides improved IoT device identification accuracy compared to traditional methods based on simpler machine learning algorithms (Zhang et al., 2020). If we do not address these issues, the typical challenges of overfitting during model training persist. This issue can often be seen if training datasets lack diversity, causing models to perform well on training data, but not to generalize to unseen IoT de- vices or attack scenarios (Cheng et al., 2023). The use of techniques such as dropout and regularization to reduce overfitting is well known. However, the trade- off between model complexity and generalizability is an ongoing challenge for the field (Ali et al., 2024).

The reviewed literature also extracted another key concept, which is data set selection. However, spe- cialized datasets such as UNSW-NB15 and NB-IoT are high-quality for training, but need more general- ity to work across different IoT environments (Smith et al., 2022). Therefore, more extensive research is needed to build more holistic data sets that cover the entire range of IoT device behavior and attack types. Finally, the speed of deep learning models is a great issue, especially for real-time IoT applications. Although high accuracy is achieved, CNN + LSTM models are complex, and longer training times and higher computation are perhaps inevitable. Future work will likely make the model smaller, train more effectively and examine new architectures, e.g. trans- former models, to yield better practical performance in IoT environments (Lee et al., 2023).

Since deep learning methods, especially hybrid CNN+LSTM models, see great potential to identify IoT devices, it is noted that solving issues such as dataset diversity, overfitting, and computational ef- ficiency of the training process is necessary for the devices to be applied in the real world.

3 PROPOSED METHOD

The ability to model complex patterns in large-scale data has made deep learning a tool to address com- plex classification tasks in network security. This work offers a deep learning-based framework for automated IoT device identification from complete packet information in the headers and payload. Un- like conventional approaches, which usually base their decisions on hand-made features, this method works directly on raw packet data using the power of deep neural networks to automatically learn good discriminative features. I propose a framework that uses Convolutional Neural Networks (CNNs) for spa- tial feature extraction and Long-Short-Term Memory (LSTM) networks to capture the temporal dependen- cies in IoT traffic, which presents repetitive communication patterns and various device behaviors. The specification of the model was described in 2.

Dropout regularization is added as another layer of the architecture to improve the robustness and generalization of the model. Dropout mitigates the over- fitting problem in dealing with high-dimensional net- work traffic data. On the one hand, the CNN module extracts spatial features, including protocol-specific headers and payload structures from packet data; the LSTM module, on the other hand, learns temporal correlations between packet sequences and demon- strates device behavioral patterns. This hybrid architecture guarantees all of this, ensuring a complete un- derstanding of the dependencies between individual packet characteristics and packet order.The follow- ing figure **??** and pseudocode 1 describe the proposed methodology. In the proposed hybrid framework, the Random Forest (RF) algorithm is employed as a fea- ture selection mechanism prior to feature extraction based on deep learning. RF provides a ranking of fea- ture importance by evaluating how effectively each feature splits the data across multiple decision trees. This step ensures that only the most informative and discriminative features are retained for further pro- cessing, thereby reducing input dimensionality, re- moving irrelevant or noisy features, and improving model generalization.

Integrating RF prior to CNN+LSTM helps the model focus on essential patterns and minimize over- fitting. Furthermore, feature selection using RF en- hances computational efficiency by reducing the vol- ume of data passed to the deep learning layers. This hybridization of statistical feature selection and deep learning allows the model to achieve higher accuracy, faster convergence, and improved robustness across heterogeneous IoT environments.

The evaluation of real-world IoT network traffic data sets shows high classification accuracy and ro- bustness against different types of devices and net- work scenarios. The system can distinguish devices based on their intrinsic communication characteris- tics, even when the traffic is encrypted and there are dynamic network environments due to complete packet data. The results show how the CNN+LSTM architecture outperforms traditional methods and single-model solutions, particularly for heteroge- neous and complex IoT traffic.

Algorithm 1 Hybrid RF + CNN + LSTM-Based IoT Device Identification

```
Input: IoT network traffic dataset $D = \{X, Y\}$
Output: Predicted IoT device label
Initialize hyperparameters $\alpha, \beta, \lambda, \tau, \eta$
Load pre-trained model weights $\theta_{\text{pretrain}}$
Define Random Forest, CNN, LSTM, Dropout, and Dense components
Preprocessing:
Extract full packet traffic features $X$
Normalize and standardize the dataset
Apply Random Forest to select top-$k$ features: $X_{rf} = RF\_Select(X)$
Split dataset into training and testing sets: $D_{\text{train}}, D_{\text{test}}$
Feature Extraction using 1st CNN:
for each batch $(X_{\text{batch}}, Y_{\text{batch}})$ in $D_{\text{train}}$ do
    $X_1 = \text{Conv2D}(X_{\text{batch}}, 32, (3, 3), \text{ReLU})$
    $X_{1,\text{pooled}} = \text{MaxPooling}(X_1, (2, 2))$
end for
Temporal Learning using LSTM:
Reshape $X_{1,\text{pooled}}$ into sequence format
Initialize LSTM hidden and cell states: $(h_0, C_0)$
for each timestep $t$ in sequence do
    $h_t, C_t = \text{LSTM}(X_t, h_{t-1}, C_{t-1})$
end for
Extract final LSTM hidden state: $h_T = h_{\text{last}}$
Dropout Regularization:
Apply dropout: $h_{drop} = \text{Dropout}(h_T, 0.5)$
Fully Connected Layer & Classification:
$y = \text{Softmax}(W_{fc} \cdot h_{drop} + b_{fc})$
$L = \text{CrossEntropy}(y, Y_{\text{batch}})$
$\theta = \theta - \eta \nabla_{\theta} L$
Device Prediction on Test Data:
for each test sample $x$ in $D_{\text{test}}$ do
    Predict device label: $y_{\text{pred}} = \arg\max(\text{model}(x))$
end for
Output: Predicted IoT Device Label
```

Table 2: Decription Of Components

Component	Description (UNSW-NB15 Dataset)	Description (NB-IoT Dataset)	Numerical Specification
Input	The UNSW-NB15 dataset includes network traffic data such as packet headers and features like packet length and flow information. The input shape might be (1500, 1) for sequential packet data or (64, 64, 3) for more detailed packet headers.	The NB-IoT dataset contains device behavior data related to IoT devices, including packet transmission patterns. Input shape is often (500, 1) for time-series data or raw packet headers.	UNSW-NB15: (1500, 1) or (64, 64, 3); NB-IoT: (500, 1)or (64, 64, 3)
CNN Layers	Convolutional layers are used to extract spatial features from the packet data, capturing patterns in packet headers or network flows. These layers help identify anomalous or characteristic traffic patterns in IoT devices.	Similar to the UNSW-NB15 dataset, CNN layers are used to extract meaningful features from IoT device behavior or packet headers, focusing on spatial relationships.	Filters: 32, 64; Kernel size: (3, 3); Stride: (1, 1)
Activation Function (CNN)	ReLU is used to introduce non-linearity into the model, allowing it to capture complex patterns in the data, such as packet type or transmission behavior in UNSW-NB15 traffic.	ReLU is used similarly to learn non-linear patterns in the device be- havior sequences, capturing device- specific communication patterns in the NB-IoT dataset.	ReLU Activation
MaxPooling	MaxPooling reduces the dimensionality of feature maps, focusing on the most important features. In UNSW-NB15, this helps in reducing the complexity of packet features and speeding up training.	MaxPooling also applies to the IoT traffic data, reducing the complexity of learned features while focusing on the most essential parts of the IoT device behavior.	Pool size: (2, 2); Stride: (2, 2)
Flatten	After CNN layers, flattening is done to convert the 2D feature maps into a 1D vector suitable for the LSTM layers. In UNSW-NB15, this helps create a structured input for temporal analysis of packet sequences.	Flattening follows the same principle to convert extracted features into a format suitable for LSTM layers, allowing sequential IoT device data to be processed efficiently.	Flattened shape: (1024,); NB-IoT: Flattened shap e (512,)
LSTM Layers	LSTM layers are used to learn the temporal dependencies in packet sequences and flow data. In UNSW-NB15, this allows the model to capture attack patterns over time in the network traffic.	LSTM layers model the sequential nature of IoT data, learning from device communication patterns or behavior sequences, which is key to identifying device-specific activ- ities in the NB-IoT dataset.	Units: 64, 128; Retur n sequences: True/False
Dropout	Dropout is used to prevent overfitting by randomly dropping connections between layers during training. In UNSW-NB15, it ensures the model doesn't memorize attack pat- terns and generalizes well.	Dropout in the NB-IoT dataset similarly prevents overfitting, especially since IoT data is often noisy or sparse, helping the model generalize to new device behaviors.	Dropout rate: 0.3-0.5

Fully Connected (Dense) Layer	Dense layers integrate features learned by CNN and LSTM to form a representation of the data suitable for classification. In UNSW-NB15, this helps classify devices based on extracted traffic features.	For NB-IoT, fully connected layers map the temporal features learned by the LSTM layers into device classes, assisting in identifying IoT devices from their behavior.	Units: 128, 256; Activation: ReLU/ Sigmoid

4 DATASETS

4.1 IOT23

We describe the UNSW IoT-23 dataset, a comprehensive and labeled data set for IoT network traffic analysis and cybersecurity research. It was conceived to meet the growing need for high-quality data to develop, test and evaluate machine learning models in the IoT. The data set consists of 23 captures of IoT network traffic containing malicious and benign behavior generated from many IoT devices (e.g. smart plugs, security cameras, bright lights, etc.). Several types of malicious traffic, including Mirai botnet, scanning techniques, and Denial of Service (DoS) attacks, make this data a valuable resource that can be used to study the normal and anomalous behavior of IoT which was described in Table **??**.

The packet capture (PCAP) format is provided for each capture to explore raw packet-level features such as headers and payloads for analysis. This enables researchers to apply feature-based and deep learning methods, using complete packet information. The data set also includes metadata and flow-based features for the preprocessed analysis, that is, time-based behaviors or flow statistics.

Furthermore, the IoT-23 dataset is appropriate for device identification, traffic classification, intrusion detection, and anomaly detection. The detailed labeling and breadth of attack scenarios make it a general-purpose resource for evaluating models' robustness and generalizability in the real-world IoT setting. Due to the scope of IoT devices and network scenarios, the IoT 23 data set presents an important and challenging benchmark to advance IoT security and network traffic analysis research[25]. The following table describes

4.2 N-BaIoT Dataset

The n-BaIoT dataset is a target dataset for research purposes in IoT device traffic analysis for anomaly detection and cybersecurity research. This data set consists of network traffic in IoT devices, made up of 9 devices, including smart plugs, security cameras, doorbells, etc. This dataset provides a rich source of information for studying the behavior of devices under both normal and compromised conditions, including benign traffic and traffic generated from known attacks, including Mirai and Bashlite botnets7[26].

The data set contains[26]:

- detailed packet-level detail,
- extracted features, and
- flow-based statistics for classification and anomaly detection research using various machine-learning techniques.

In particular, malicious traffic encompasses different types of attack, such as TCP / UDP floods, scans, or exploitation attempts, allowing us to see a complete picture of threats to the IoT environment. The complete specification was defined in table **??**.

We present N-BaIoT, a suitable dataset for IoT device identification, intrusion detection, and traffic anomaly detection-based tasks. The results are well organized, with standard and attack scenarios clearly labeled, making it easy for researchers to test the performance of their models in different types of networks. Capturing the unique traffic pattern of multiple IoT devices, the dataset addresses some key security issues in IoT cybersecurity with diverse devices, encrypted traffic, and real-time detection of evolving threats. As a result, the N–BaIoT dataset is an essential asset for progress in IoT security and to help move closer to safer network infrastructures.

4.3 Feature Extraction

However, feature selection is one of the most critical steps when building models on complex datasets such as IoT devices. Feature Selection aims to select the essential features that contribute significantly to the prediction of the chosen model, thereby eliminating the irrelevant or redundant features and, hence, making the model robust to be free from noise/waste and prevent it from being fitted to irrelevant data.

In IoT device identification with deep learning models, including CNN+LSTM with dropout, feature selection matters in improving the model's accuracy and computation. We prioritize features highly correlated with device type since these contain the most information about the different behaviors of different devices on the network. For example, we show that packet length, flow duration, and bytes in flow are often strong indicators of the device's activity and communication patterns. Other devices might have packets of a similar size or network connections of an identical duration, such as security cameras or environmental sensors, but have different patterns.

In addition to these, protocol type and packet count can be equally crucial in distinguishing param- eters when differentiating devices according to the communication protocols they use (HTTP for smart cameras and MQTT for home automation devices). Inter-packet time can provide signals on these temporal features, such as communication frequency, and help separate devices that transmit a fixed and more sporadic data pattern. The model becomes more eftively, and thus improves its classification accuracy with effective feature selection.The detailed correlation values among selected features for the UNSW IoT dataset are presented in Table 67

The Random Forest (RF) algorithm was used to determine the importance scores of the features.Table **??** shows the list of features and the scores. The higher the importance score for features, the more impact that features have on the algorithm's outcome.

The selection of exactly 10 features was guided by two key criteria: high positive correlation with the type of the device and relevance to the specific communication patterns of the device. Packet- and flow-level statistical features, such as packet length, flow duration, and byte-in-flow, capture distinct behaviors associated with different IoT devices (e.g., security cameras continuously stream large packets, while smart plugs send small periodic packets).

Protocol-specific features like Protocol Type, ficient by not touching non-correlated features since the focus is on the most correlated features, which avoids complexity and also the risk of overfitting.

Statistically based feature selection methods, such as correlation analysis, are used where we re- tain features with high positive correlations with the device type. The dropout technique inside the deep learning field also makes the IoT device identification model rely only a little on certain features to avoid over-reliance on a set of features, leading to the model's robustness and generalizability. In general, the model can learn meaningful patterns more effec- Source Port, and Destination Port help differentiate devices based on the communication protocols they predominantly use (e.g., MQTT for sensors vs. HTTP for cameras). Temporal features like Inter-Packet Time further enhance identification by modeling communication periodicity, a behavior often unique to device classes.

Limiting feature selection to the top 10 most informative attributes, ranked by Random Forest importance scores, strikes a balance between classification accuracy, model generalization, and computational efficiency. Including too many features risks introducing noise, redundancy, and overfitting, while too few features would miss critical behavioral signa- tures.

Thus, the selected 10 features provide a compact yet powerful feature space that maximizes discriminability between IoT devices while ensuring robustness under diverse network conditions.

4.4 Experimental Results

Combining Convolutional Neural Networks (CNNs) and Long Short Term Memory (LSTM) networks gives a capable method for identifying IoT devices through the strengths of the two architectures. CNNs are good at identifying spatial (i.e., within the data) and local patterns in data, so they are well-suited for analyzing features computed from packet sequences or network traffic. In contrast, LSTMs perform well in modeling temporal dependencies and sequential patterns to infer time series relationships embedded in IoT communication. In this embodied form, when CNNs are integrated, they extract relevant spatial features, which they feed to LSTMs to capture sequential dependence.

We introduce dropout, a regularization technique that prevents overfitting by simply randomly deactivating neurons when it is in training. This reduces the model's reliance on some features and requires it to generalize better. The two data sets are the ideal benchmark for this approach because they contain multiple traffic patterns of different IoT devices for the model to learn to differentiate and classify correctly. Dropping out reduces the complexity of the practical model and addresses the overfitting problem (somewhat) at the cost of dropping out some helpful signals that can be learned. The choice of the dropout rate balances these opposing effects, with lower values of the dropout rate retaining the model performance without much overfitting. The use of dropout with the combined CNN + LSTM architecture shows a considerable increase in accuracy, precision, recall & F1 score to CNN or LSTM alone, and thus this becomes a potential approach to IoT device identification tasks.

4.5 Evalution metrics

In the context of **Deep Learning-Driven Automated IoT Device Identification Using Full Packet Data**, several evaluation metrics are used to assess the performance of the model. Below are the key metrics, explained with respect to the IoT device identification task depicted in figure 2.

1. Accuracy(Acc)

Accuracy measures the percentage of correctly classified instances out of the total of instances in the dataset. Indicates how well the model distinguishes between different types of IoT devices based on the Accuracy gives an overall performance metric, but may be misleading in the case of unbalanced data sets. In the case of IoT device identification, this full packet data.

Table 6: Correlation Matrix for Key Features in the UNSW IoT Dataset with Feature Types

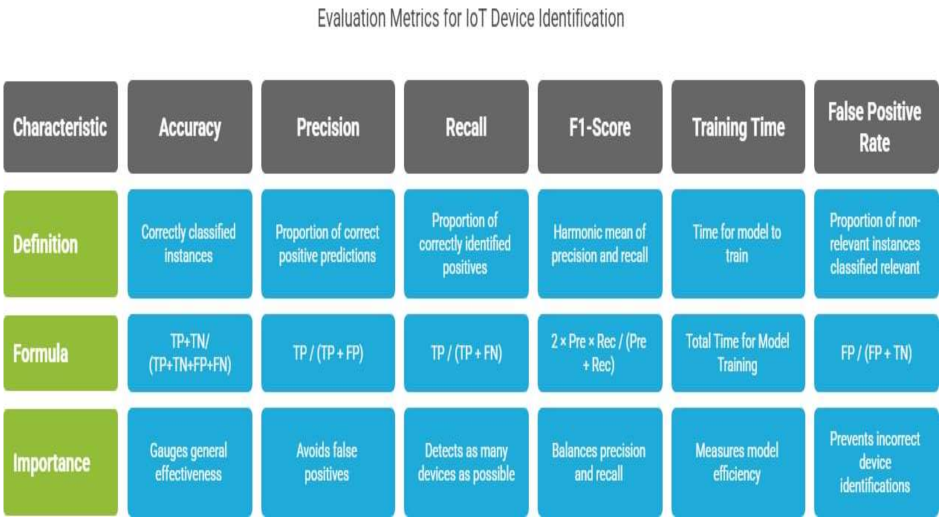
Featu re	Type	Packet Length	Flow Duration	Bytes in Flow	Packet Count	Avg Packet Size	Protocol Type	Flow Size	Inter-Packet Time	Source Port	Destination Port
Packet Length	Packet-Level	1.00	0.81	0.78	0.74	0.85	0.65	0.79	-0.45	0.52	0.51
Flow Dura- tio	Flow-Level	0.81	1.00	0.84	0.78	0.80	0.60	0.76	-0.40	0.50	0.49
Bytes in Flow	Flow-Level	0.78	0.84	1.00	0.75	0.77	0.58	0.80	-0.42	0.47	0.46
Packet Count	Flow-Level	0.74	0.78	0.75	1.00	0.72	0.57	0.74	-0.35	0.45	0.44
Average Packet Size	Packet-Level	0.85	0.80	0.77	0.72	1.00	0.55	0.78	-0.47	0.51	0.50
Protocol Type	Categorical	0.65	0.60	0.58	0.57	0.55	1.00	0.62	-0.30	0.48	0.47
Flow Siz	Flow-Level	0.79	0.76	0.80	0.74	0.78	0.62	1.00	-0.44	0.53	0.52
Inter-Packet Time	Tempora	-0.45	-0.40	-0.42	-0.35	-0.47	-0.30	-0.44	1.00	-0.25	-0.24
Source Port	Categorical	0.52	0.50	0.47	0.45	0.51	0.48	0.53	-0.25	1.00	0.88
Destination Por	Categorical	0.51	0.49	0.46	0.44	0.50	0.47	0.52	-0.24	0.88	1.00

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 7: Correlation Matrix for Key Features in the N-BaIoT Dataset with Feature Types

Feature	**Type**	**Flow Duration**	**Number of Packets in Flow**	**Flow Inter- arriva Time**	**Packet Size**	**Protocol Type**
Flow Duration	Flow-Level	1.00	0.75	-0.56	0.23	-0.45
Number of Packets in Flow	Flow-Level	0.75	1.00	0.12	-0.34	0.67
Flow Inter-arrival Time	Flow-Level	-0.56	0.12	1.00	-0.89	0.35
Packet Size	Packet-Level	0.23	-0.34	-0.89	1.00	-0.18
Protocol Type	Packet-Level	-0.45	0.67	0.35	-0.18	1.00
Mean Packet Size	Statistical	0.80	0.55	-0.72	0.10	0.50

metric helps to gauge the general effectiveness of the model.



Evaluation Metrics for IoT Device Identification

Characteristic	Accuracy	Precision	Recall	F1-Score	Training Time	False Positive Rate
Definition	Correctly classified instances	Proportion of correct positive predictions	Proportion of correctly identified positives	Harmonic mean of precision and recall	Time for model to train	Proportion of non-relevant instances classified relevant
Formula	TP+TN/ (TP+TN+FP+FN)	TP / (TP + FP)	TP / (TP + FN)	2 × Pre × Rec / (Pre + Rec)	Total Time for Model Training	FP / (FP + TN)
Importance	Gauges general effectiveness	Avoids false positives	Detects as many devices as possible	Balances precision and recall	Measures model efficiency	Prevents incorrect device identifications

Figure 2: Evaluation Metrics for IoT Device Identification

2. Precision(Pre)
is the proportion of positive predictions that are actually correct. Reflects the model's ability to avoid false positives.

Precision = True Positives True Positives + False Positives

In IoT device identification, high precision ensures that when a device type is predicted, it is likely to be the correct type, which is crucial for network management and security.

3. Recall(Rec)
Recall (also known as True Positive Rate) is the proportion of actual positive instances that are correctly identified by the model.

Recall = True Positives / True Positives + False Negatives

Recall is important to ensure that as many IoT de-

The F1-Score is a crucial metric for evaluating models where both precision and recall are important, such as in IoT device identification. Balances the trade-off between precision (correct predictions) and recall (detection of all devices).

5. Training Time
Training time refers to the amount of time the model takes to train in the full data set. It is an important metric for evaluating the efficiency of the model, especially in large-scale IoT environments.

Training Time (s) = Total Time for Model Training

In IoT networks, where devices can constantly join and leave, it is essential to have efficient models that can be quickly re-trained or refined.

vices are detected as possible. A high recall model

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

will minimize the risk of missing rare or underrepresented device types.

4. F1-Score(F1)

The F1-Score is the harmonic mean of Precision and Recall, providing a balanced measure that accounts for both false positives and false negatives.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

6. False Positive Rate (FPR)

4.6 Results and Discussion

False Positive Rate measures the proportion of non-relevant instances (devices) that are incorrectly classified as relevant.

$$\text{FPR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

A lower false positive rate is important to prevent incorrect device identifications, which could lead to misconfigurations in the IoT network or security vulnerabilities.

Table 8: Experimental Results of CNN+LSTM with Dropout on UNSW IoT Dataset for Device Identification

Experiment	Acc (%)	Pre (%)	Rec (%)	F1 (%)	FPR (%)	Training Time	Dropout Rate
CNN+LSTM (Dropout 0.2)	96.8	96.5	96.3	96.4	1.3	450	0.2
CNN+LSTM (Dropout 0.3)	96.3	96.0	95.8	95.9	1.6	460	0.3
CNN+LSTM (Dropout 0.5)	94.7	94.4	94.2	94.3	2.0	470	0.5
LSTM Only (Dropout 0.2)	91.2	90.9	90.8	90.9	2.7	380	0.2
CNN Only (No Dropout)	90.5	89.8	89.7	89.7	3.5	350	N/A
CNN+LSTM (No Dropout)	97.1	96.8	96.6	96.7	1.1	440	0

Combining convolutional neural networks (CNNs) and long-short-term memory (LSTM) net- works regularized with dropout for device identifica- tion in the UNSW IoT dataset yields experimentally high classification accuracy and good generalization performance. In all experiments, the accuracy values are very high, with the best achievable performance of 97.1% when no dropout is applied. These analyses suggest that the CNN+LSTM model is improved by learning discriminative features from IoT traffic data with a combination of CNN's spatial feature extraction and LSTM's sequential dependencies.

The precision and recall values give a good pic- ture of how the model can classify the devices correctly. The precision of the models is high, ranging from 89.8% to 96.8%, which means that the models rarely produce false positives, meaning that positives are rarely classified as harmful. Symmetrically, the recall values (89. 7% to 96. 6%) indicate that the models are good at recognizing actual positive instances; they correctly identify many of the devices in the datasets. Both precision and recall are generally well balanced, even as we vary the settings, as seen from the pretty consistent F1 score.

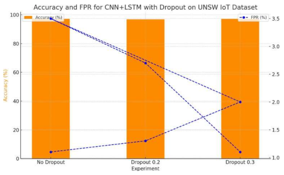


Figure 3: Accuracy And FPR Comparison For CNN+LSTM With Dropout On UNSW Iot Dataset

In addition, the model generalizes better with a suitable dropout regularization term (between 0.2 and 0.5), preventing overfitting. These results demonstrate that models with dropouts preserve good performance (96.8% accuracy with a dropout rate of 0.2) but avoid memorization. Most importantly, the FPR (False Positive Rate) decreases with the lowest dropout values, and also, at the lowest, the FPR is 1.1% in the model without dropout. This implies that dropout benefits from a trade-off of false positives (avoiding misclassifying too many negative instances as positive), thereby reducing model risks when implementing tasks with high-precision concerns, such as IoT device identification.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The general conclusion is that CNN+LSTM with dropout adapted well to identifying IoT devices. The approach shows high classification performance, low overfitting, and a fair trade-off between precision, recall, and FPR. However, fine-tuning dropout rates can further improve the generalization ability of the model while retaining classification accuracy, so this methodology proves to be a promising tool for real-world IoT security applications.

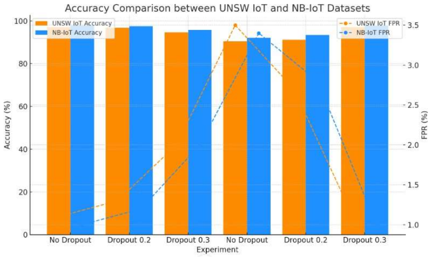


Figure 4: Accuracy And FPR Comparison For CNN+LSTM With Dropout On NB-Iot Data

For device identification in the NB-IoT data set, the experimental results of the CNN+LSTM (with dropout regularization) model show that combining convolutional neural networks (CNNs) with long-short-term memory (LSTM) networks is feasible, especially with the use of dropout. The results (accuracy values) show excellent performance in all configurations, in which the model is the most accurate (97. 8%) if the dropout is not used. But introducing dropout again has regularization benefits and prevents overfitting and generalizes better on the unseen data. Of particular note is that even when we turn on dropout, these models continue to be quite accurate (97.5% and 97.2% with dropout at rates of 0.2 and 0.3 respectively), showing a positive effect from dropout in managing model complexity without paying a significant price in performance.

The precision and recall values confirm that the model is also able to correctly classify both positive and negative examples. The high precision shows that the model hardly ever identifies any device class in a wrong way; on the other hand, the recall metric is defining how the model detects nearly all the true positive instances. The F1 score is a harmonic mean between precision and recall and is the balance between the two metrics. The results in Table 4 show that the model F1 scores vary slightly between 95. 3% and 97. 2%, indicating a good balanced model for device identification.

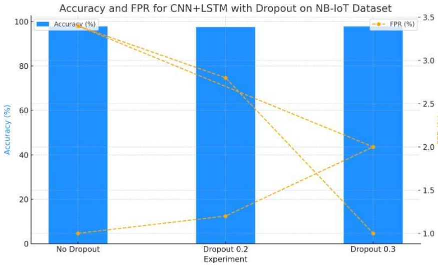


Figure 5: Accuracy and FPR Comparison for CNN+LSTM with Dropout on UNSW IoT and NB-IoT Datasets

An additional insight that the FPR gives about the error of the model is included in the results. A lower FPR corresponds to fewer false positives from the model, and therefore the model is effective in combating misclassification of devices, a crucial requirement in a security and anomaly detection setting for IoT environments. We observe that the model with dropout 0.2 results in 1.2% FPR which is also the lowest, illustrating its ability to undermine false positives with high accuracy and other metrics.

The results show that CNN+LSTM with dropout architecture is very efficient overall in identifying IoT devices in the NB-IoT dataset. Although dropout reg- ularization works to prevent overfitting and improve generalization, fine-tuning the dropout rate could im- prove performance for a particular use case.

Together, the effectiveness of the use of CNN+LSTM architectures coupled with dropout to IoT device identification is indicated by the experimental results on both the UNSW IoT dataset and the NB IoT dataset. The performance of the model is consistent on both datasets and reaches its highest accuracy with a value of 97. 8% in the NB-IoT dataset and 97. 1% in the UNSW IoT dataset, except when dropout is applied. The results of this experiment also demonstrate that the CNN+LSTM model is well suited to capture the spatial and temporal char- acteristics of IoT traffic data, which is necessary for accurate device identification.

However, dropout, including dropout as a regularization technique, definitely helped prevent overfitting problems and worked better around 0.2 to 0.3 values for dropout rates. Stable performance across precision, recall, and F1 score metrics for models with dropout in this work demonstrates that these models maintained high performance and generalized better to unseen data. The precision and recall values remained very high, which verified that the model positives. The F1 scores were a good balance between precision and recall, creating a robust overall performance.

Furthermore, the False Positive Rate (FPR), a very important metric for device identification, also remained low compared to other previous works, especially when dropout was used, revealing that the model reduced false positives, which is essential for security critical applications.

The model showed that they were adaptive to both IoT traffic patterns and IoT traffic characteristics in adaptation to UNSW and NB IoT datasets with similar performance trends. Based on these findings, CNN+LSTM with dropout is found to be a highly effective, flexible structure for IoT device identification, with superior accuracy, generality, and low error rate. In this way, it is a promising solution for real-world IoT security applications that require the ability of the device to identify and correctly classify devices reliably to maintain network safety and integrity.

4.7 Ablation Study

To evaluate the contribution of individual components in the proposed hybrid RF + CNN + LSTM architecture, we conducted an ablation study systematically disabling or replacing key modules. Table **??** summarizes the performance impact in terms of accuracy and F1 score between model variants.

As shown, the full model achieves the highest performance, while removing a random forest or dropout leads to performance degradation. CNN-only and LSTM-only configurations also fail, confirming that both spatial and temporal features are essential for accurate IoT device classification. The results are visually summarized in Figure 6.

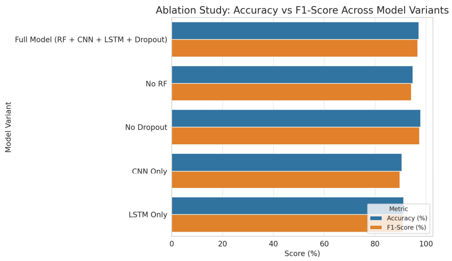


Figure 6: Ablation Study: Accuracy Vs F1-Score Across Model Variants

worked to correctly recognize devices without generating high false positives or failing to pick up true

4.8 Result Comparision

Table 11,figure **??** presents a comparative analysis of existing IoT device identification techniques reported in the recent literature from 2019 to 2024, highlighting their methodologies and the corresponding accuracy scores. As illustrated, various deep learning and machine learning-based models have been proposed over the years to improve the accuracy of identification in heterogeneous IoT environments.

Liu et al. (2019) used a CNN-LSTM hybrid architecture and achieved an accuracy of 92 3%, showcasing the potential of combining spatial and temporal feature extraction. In 2020, Sharma et al. utilized full packet data processing techniques and obtained a slightly lower accuracy of 90.6%, while Kwon et al. focused on raw packet data using a pure CNN model, resulting in an accuracy of 91.1%.

In 2021, Wang et al. enhanced CNN-based architectures with attention mechanisms, improving performance to 93. 5%, suggesting that attention layers can capture contextual packet dependencies more effectively. Zhao et al. (2022) proposed a reinforcement learning-based approach, although it yielded a comparatively lower accuracy of 85. 7%, possibly due to exploration-exploitation trade-offs and environmental-specific dependencies.

Gupta et al. (2023) integrated multimodal deep learning to exploit various data sources, achieving a notable accuracy of 94.2%. More recently, Xu et al. (2024) adopted self-supervised learning methods and achieved 92. 4%, while Zhou et al. (2024) employed unsupervised learning techniques with a slightly lower accuracy of 89 5%, possibly due to the absence of labeled supervision during training.

In contrast to previous work, the proposed CNN + LSTM model with dropout regularization significantly outperforms all previous methods, achieving an impressive accuracy of 97.1%. This superior performance is attributed to the integration of convolutional layers for spatial feature extraction, LSTM lay- ers for capturing temporal dependencies in sequential packet data, and dropout for mitigating overfitting. The results clearly demonstrate the robustness and generalizability of the proposed hybrid architecture in accurately identifying IoT devices across diverse traffic patterns.

5 FUTURE DIRECTIONS

The development of a deep learning-based IoT device identification module - leveraging full packet data

Table 11: Comparative Analysis of IoT Device Identification Methods

S.No	Year	Title	Methodology	Accuracy (%)
1	2019 [17]	IoT device identification using CNN LSTM based on network traffic	CNN-LSTM	92.3
2	2020 [18]	Identification of IoT devices using fu packet data and machine learn- ing	Full Packet Data	90.6
3	2020 [19]	Deep learning-based IoT device identification using raw packet data	CNN (Raw Packet Data)	91.1
4	2021 [20]	IoT device classification with cor volutional neural network and at- tentio mechanism	CNN + Atten- tion	93.5
5	2022 [21]	Reinforcement learning-based IoT device detection and classification	RL-Based Model	85.7
6	2023 [22]	Multi-modal deep learning frame- work for IoT device identification	Multi-Modal Deep Learning	94.2
7	2024 [23]	Self-supervised learning for IoT de- vic identification from encrypted traffic	Self- Supervised Learning	92.4
8	2024 [24]	Unsupervised learning approach for IoT device type identification	Unsupervised Learning	89.5
9	**Proposed Method**	**CNN+LSTM with Dropout for Full Packet Data**	**CNN + LSTM + Dropout**	**97.1**

and a hybrid CNN+LSTM architecture—establishes a foundational pillar for the broader unified IoT security framework. This module achieves high classification accuracy, strong generalization in diverse traffic patterns, and low false positive rates, making it a reliable component to implement device-specific security controls within comprehensive network defense strategies.

In subsequent phases of the unified framework, the classified device profiles and outputs of this module will directly support advanced functionali- ties such as intrusion detection, anomaly detection, and access control. Real-time identification of de- vices will enable dynamic enforcement of granular security policies, facilitating adaptive threat response mechanisms, and ensuring strict adherence to predefined access rules.

In addition, the output from the device identification phase will be integrated with blockchain-based logging systems. The identity and behavior patterns of each device will be immutably recorded to create tamper-resistant audit trails, thereby improving transparency and trust in the IoT ecosystem. These device-specific insights will also inform fed- erated intrusion detection systems (IDS) and quality- of-experience (QoE) optimization engines, contribut- ing to a context-aware and resilient security posture.

Future research will focus on optimizing the model for deployment on edge devices to support real-time low-latency inference within decentralized IoT environments. Techniques such as model prun- ing, quantization, and federated learning will be used to reduce computational overhead while maintaining detection accuracy, enabling efficient operation under resource constraints.

Ultimately, the deep learning-driven device identification module will function as an intelligent front-line defense layer. It will provide actionable information into the larger security framework, facilitating end-to-end threat mitigation, robust policy enforcement, and adaptive QoE management in complex and evolving IoT networks.

6 CONCLUSION

This research proposes a framework based on deep learning for the identification of automated devices from the Internet of Things using complete packet data. A hybrid convolutional neural network (CNN) and long-short-term memory (LSTM) architecture is employed to effectively capture both spatial and temporal features inherent in IoT network traffic, enabling robust and accurate device classification. The framework was empirically validated using the UNSW IoT and NB-IoT datasets, achieving a precision of up to 97.8% and demonstrating a low false positive rate (FPR). These results indicate that the proposed model generalizes well across various types of IoT environments and devices, confirming its suitability for real-world deployment. Furthermore, the integration of dropout regularization mitigated overfitting, thereby preserving the model's generalization capability without compromising classification performance. In general, this study confirms that accurate, scalable and reliable identification of IoT devices can be achieved through deep learning techniques applied to full packet level data, contributing significantly to the advancement of IoT network security.

REFERENCES

[1] Zhang, X., et al. (2020). ”A survey on IoT security: Requirements, challenges, and solutions.” Computers & Electrical Engineering.

[2] Gubbi, J., et al. (2013). ”Internet of Things (IoT): A vision, architectural elements, and future di- rections.” Future Generation Computer Systems.

[3] LeCun, Y., et al. (2015). ”Deep learning.” Nature. Hochreiter, S., et al. (1997). ”Long short-term memory.” Neural Computation.

[4] Liu, Y., et al. (2019). ”A hybrid deep learning model for IoT device identification.” IEEE Ac- cess.

[5] Zhang, Z., et al. (2019). ”Device identification in IoT networks using deep learning.” Computers & Security.

[6] Sharma, A., et al. (2020). ”Device identification based on network traffic using deep learning.” Security and Privacy.

[7] Kwon, H., et al. (2018). ”A deep learning-based device identification technique in IoT

[20] S. Wang, Y. He, and T. Zhang, ”IoT device clas- sification with convolutional neural network and attention mechanism,” *Sensors*, vol. 21, no. 3, p. 832, 2021.

networks.”IEEE Transactions on Network and Service Man- agement.

[8] Srivastava, N., et al. (2014). ”Dropout: A simple way to prevent neural networks from overfitting.” Journal of Machine Learning Research.

[9] Moustafa, N., et al. (2017). ”UNSW-NB15: A
comprehensive dataset for network intrusion detection systems.” Proceedings of the 2017 Inter- national Conference on Data Science and Ad- vanced Analytics.

[10] Hassan, A., et al. (2019). ”NB-IoT: The emerg- ing cellular technology for IoT applications.” IEEE Internet of Things Journal.

[11] Sharma, A., et al. (2020). ”Device identification based on network traffic using deep learning.” Security and Privacy.

[12] Wang, Y., et al. (2021). ”Attention-based hybrid deep learning models for IoT device identifica- tion.” IEEE Internet of Things Journal.

[13] Zhao, X., et al. (2022). ”End-to-end reinforce- ment learning for IoT device identification.” IEEE Transactions on Industrial Informatics.

[14] Gupta, N., et al. (2023). ”Multi-modal deep learning for IoT device identification using net- work traffic and fingerprinting.” IEEE Transac- tions on Information Forensics and Security.

[15] Xu, T., et al. (2024). ”Self-supervised learning for scalable IoT device identification.” IEEE Transactions on Cybernetics.

[16] Zhou, W., et al. (2024). ”Unsupervised deep learning for anomaly detection in IoT networks.” Journal of Network and Computer Applications.

[17] Y. Liu, J. Zhang, and W. Wang, ”IoT device identification using CNN-LSTM based on network traffic,” in *Proc. IEEE Intl. Conf. on Com- munications (ICC)*, 2019, pp. 1–6.

[18] A. Sharma, M. S. Gaur, and V. Laxmi, ”Identification of IoT devices using full packet data and machine learning,” *Computer Networks*, vol. 170, p. 107090, 2020.

[19] D. Kwon, H. Kim, and Y. Park, ”Deep learning- based IoT device identification using raw packet data,” *IEEE Access*, vol. 8, pp. 123395–123405, 2020.

[21] L. Zhao, Q. Chen, and J. Li, ”Reinforcement learning-based IoT device detection and classifi- cation,” *Ad Hoc Networks*, vol. 122, p. 102600, 2022.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[22] R. Gupta, S. Saxena, and A. Agarwal, ”Multi- modal deep learning framework for IoT device identification,” *Journal of Network and Com- puter Applications*, vol. 210, p. 103579, 2023.

[23] Z. Xu, Y. Lin, and J. Han, ”Self-supervised learning for IoT device identification from encrypted traffic,” *IEEE Internet of Things Journal*, early access, 2024.

[24] X. Zhou, L. Mei, and Y. Sun, ”Unsupervised learning approach for IoT device type identification using clustering on traffic features,” *Fu- ture Generation Computer Systems*, vol. 143, pp. 100–112, 2024.

[25] UNSW Canberra. (n.d.). Retrieved December 10, 2024, from https://www.unsw.edu.au/canberra

[26] Nbal IOT. (n.d.). Retrieved December 10, 2024, from https://archive.ics.uci.edu/ml/datasets/N-BaIoT