Arabian Journal for Science and Engineering (2022) 47:9907–9930
https://doi.org/10.1007/s13369-021-06472-z

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

A Novel Distributed Stack Ensembled Meta-Learning-Based Optimized Classification Framework for Real-time Prolific IoT Traffic Streams

Manish Snehi[1] · Abhinav Bhandari[1]

Received: 16 August 2021 / Accepted: 1 December 2021 / Published online: 18 January 2022

Abstract

The concurrence of state-of-the-art Industrial 5G, Cyber-Physical Systems, Smart-Systems, Industrial Internet of Things, and Additive Manufacturing paves the next-level digital remodeling. However, the transfiguration unwittingly tailpiece an operational onus on the smart-environment operators. The multiplicity and classes of IoT devices operating in the intelligent environment are myriad. The characterization of ingress network traffic and the accurate classification of devices is necessary to efficiently manage the devices and offer cutting-edge security solutions and quality of Service (QoS). The paper addresses these challenges by offering a novel intelligent framework for traffic classification leveraging behavioral attributes of IoT traffic. The paper's contributions to the research community are fourfold. Firstly, the paper proposes a novel IoT classification framework based on Stack-Ensemble for real-time high-volume IoT traffic. The experimental results indicate that the proposed novel Stack Ensemble model can extract the best out of base models and demonstrate an accuracy of 99.94%. The intelligent models are evaluated over multiple dimensions to project the isometric view of the model performance and the experimental results. To achieve that goal, all the performance metrics that most researchers most often miss have been elucidated. Secondly, the paper comprehends the flow-level statistical characteristics of IoT devices. Third, the paper offers the distributed, scalable, and portable framework architecture. The architecture is horizontally scalable, distributing the computational load. The framework offers an end-to-end industry-grade machine-learning pipeline and triumphs the vulnerabilities of existing solutions. Finally, the paper discusses the statistical insights into the intelligent model and the results of the experimentation study. The proposed work paves the opportunities for researchers, smart-environment operators, and developers to unfold the architecture and supplement the security solutions against cyber-attacks.

Keywords Classification · Deep learning · Docker · Distributed computing · H2O · Internet of Things · IoT · Machine learning · Stack ensemble

1 Introduction

The digital wave has acclaimed the Internet of Things (IoT) as the next unprecedented technology in a global space. The cyber-physical systems, smart eco-systems (smart homes, cities, healthcare, power grids, smart-bins), digital technologies, and enterprises are continually overhauling and embracing the IoT devices. The forecasted number of connected IoT devices is more than 25.4 billion in 2030 worldwide [16,18]. The isometry of connected IoT devices is shown in Fig. 1. The spectacular growth of IoT devices is their use in most industry verticals like electricity, gas, waste management, water supply, transportation, healthcare, storage, and government. The fastest-growing connections are M2M (Machine-to-machine), growing at the rate of 19% CAGR and are expected to grow to 14.7 billion devices by 2023. The COVID-19 pandemic has a significant impact on the growth of IoT devices due to their influence in multiple realms. The most touched areas in the IoT domain are remote patient monitoring, work from home, adoption of intelligent payment technologies, increased demand for wearable devices, crewless aerial vehicles [9,35].

The IoT devices are critical paths of the intelligent Internet of Things ecosystems and cyber-physical systems deployed at the network's edge. The consortium of the IoT devices and

✉ Manish Snehi
snehi.manish@outlook.com

Abhinav Bhandari
abhinavbhandari@pbi.ac.in

[1] Computer Science and Engineering, Punjabi University, Patiala, Punjab, India

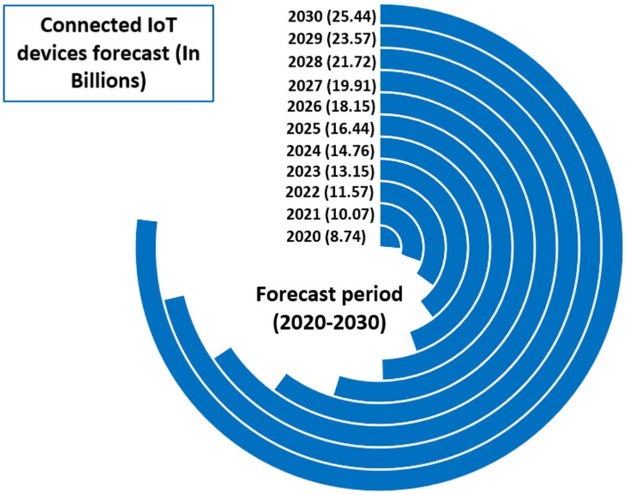


Fig. 1 Connected IoT devices forecast (2020–2030)

cyber-physical systems in the manufacturing domain has laid the Industry 4.0 and Industry 5G [26] concepts. The Industrial Revolution transformed conventionally used IoT devices into the Industrial Internet of Things (IIoT) [41]. Furthermore, IoT devices' applicability into nearly all realms has paved the way to enable efficient and sustainable systems. The substantial elevation in volume of IoT devices has compelled the instrumentation of intelligent devices into homes, hospitals, industries, cities, consumer Internet of Things, and enterprises. The accelerated multi-dimensional growth in scale creates operational, management, security, and privacy challenges [45]. The prevailing devices autonomously interact with each other and can be monitored remotely. The device administrators require insights into these devices for better management of the functions of the intelligent environment. The management process of devices is usually manual and distributed across departments. The Quality of Service (QoS) demands automated recognition, monitoring, profiling, and classification of the IoT devices to ensure the legitimate functioning ceaselessly and quarantining the machines in any cyber-attack.

1.1 Motivation

The IoT traffic might be distinctly diverse based on network specifications and available bandwidth, reaching the far horizons of the possible bandwidth spectrum. The devices may need a connection to the server continuously, periodic, sporadic, or triggered by some event. Therefore, the IoT traffic and device classification are critical to reedifying the networks with intelligent devices. The automated profiling of devices may help in traffic routing decisions, load balancing, distribution of computational load, etc. [10]. The network backbone is an essential component of an IoT ecosystem. Hence, device awareness may be crucial to boosting the system's performance.

Another enthralling motivation behind the research is that IoT devices are continuously increasing in numbers and heterogeneity. These devices always function in unison to meet a specific demand, such as Quality of Service (QoS). Given the variety of IoT devices, the market is forecasted to reach around 25.4 billion by 2030 [16,18], producing massive network traffic. The massive network traffic is fuel to big data technologies. Moreover, it is critical to process the ingress traffic in real-time for an efficient classification and security system. The amalgamation of intelligent algorithms, big data analytics, and various application areas has boosted the internet traffic classification domain.

Moreover, it is an unpleasant truth that severe challenges accompany a massive number of IoT devices. Determining the device type connected to a network enables application security measures and the assurance of the device's accurate functioning. Thorough knowledge of IoT traffic is critical for creating and optimizing an IoT traffic categorization. Indeed, the emergence of unique traffic behaviors and the widespread deployment of sensor nodes have prompted a stream of original research to offer innovative solutions.

Another compelling motive for IoT traffic profiling is for intensifying cyber-security. The IoT devices are low-power, specific-purpose devices with limited security and computational capabilities that make them vulnerable to cyber-attacks and easy to infiltrate [3,11]. The security gaps in geographically distributed, heterogeneous IoT devices have motivated the researchers to recommend defense solutions that rely on a deep understanding of normal IoT traffic behavior [54]. Relatively recent, governments are also simplifying ISP obligations concerning lawful interception (LI) of internet traffic. IoT traffic pigeonholing is an indispensable part of LI solutions [7].

1.2 Network Traffic Classification History

Researchers have invested sincere efforts in Internet traffic classification for the last two decades. After establishing the registered ports (e.g., assignment of port 80 for HTTP, port 13 for daytime, etc.) by Reynolds (RFC-1340) [38], the research in its inception employed ***port-based classification techniques*** [40]. The port-based approach used the ports registered with the Internet Assign Number Authority (IANA). The port-based method was error-prone because peer-to-peer applications used dynamic ports instead of IANA assigned ports. Following it, researchers employed the ***payload-based classification method***. The approach was processor and memory intensive and backslid for encrypted network traffic. Furthermore, a direct probe of session contents may represent a privacy violation.

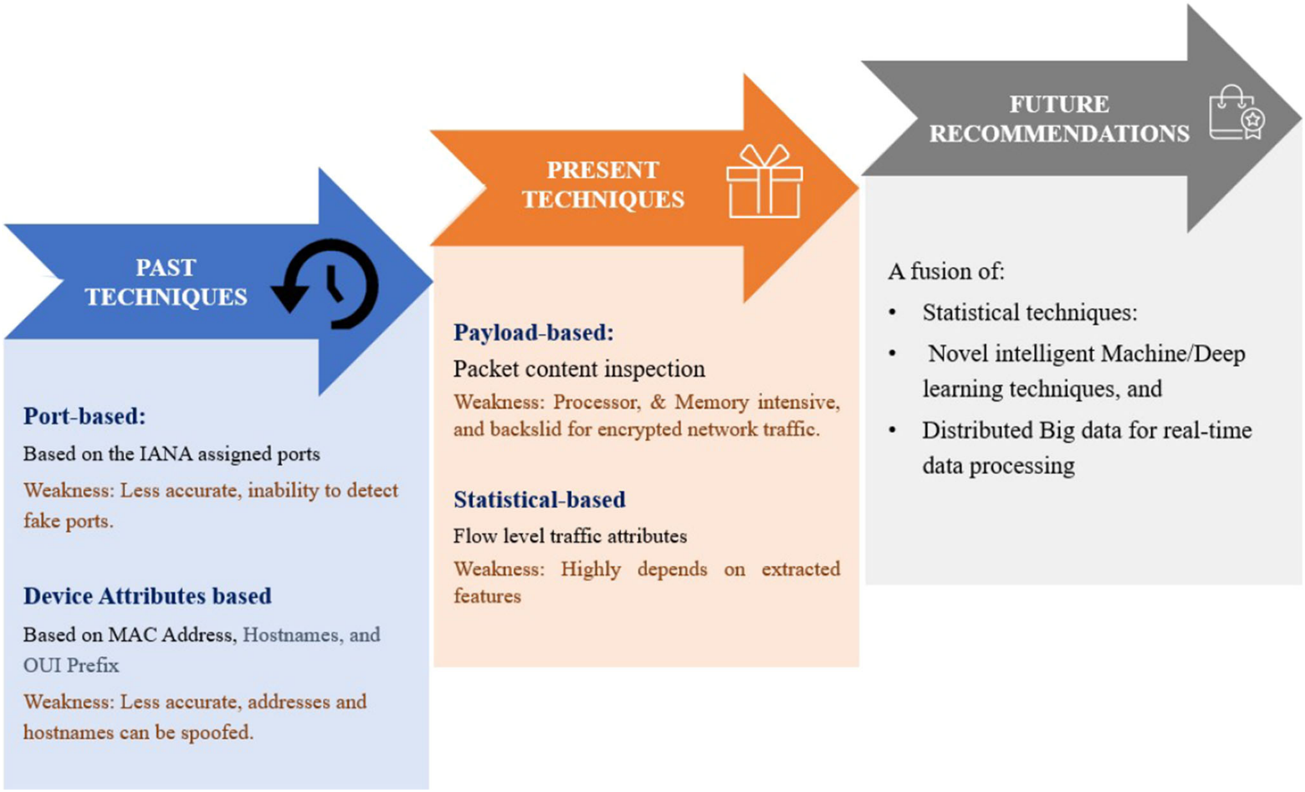

PAST TECHNIQUES
Port-based:
Based on the IANA assigned ports
Weakness: Less accurate, inability to detect fake ports.
Device Attributes based
Based on MAC Address, Hostnames, and OUI Prefix
Weakness: Less accurate, addresses and hostnames can be spoofed.
PRESENT TECHNIQUES
Payload-based:
Packet content inspection
Weakness: Processor, & Memory intensive, and backslid for encrypted network traffic.
Statistical-based
Flow level traffic attributes
Weakness: Highly depends on extracted features
FUTURE RECOMMENDATIONS
A fusion of:
• Statistical techniques:
• Novel intelligent Machine/Deep learning techniques, and
• Distributed Big data for real-time data processing

Fig. 2 Roadmap of abstract concepts for network traffic Classification

Researchers endeavoured on leveraging ***the device attributes for traffic classification***. For example, (i) Identification of devices based on Mac address and Organizationally Unique Identifier (OUI) prefix [30]. The technique is unreliable as Network Interface Cards (NIC) for IoT devices are generally manufactured by third parties. Moreover, the MAC addresses of malicious devices are spoofed. (ii) Another approach utilizes the hostname of the IoT devices for the device type identification [51]. However, hostnames are not consistent across the device families, or device users update the hostnames. Even when the manufacturers expose device names, it does not make much sense for some devices (e.g., Withing's sleep sensor has the hostname WSD-28C6).

Thenceforth, researchers proposed the ***statistical-based classification approaches*** that rely on traffic flow attributes such as flow duration, idle time, packet inter-arrival time, etc [37]. Laner *et al.* [24] demonstrated in their work that M2M device traffic could assume one of the three categories: periodic traffic exchange, traffic generated because of some event trigger, or payload exchange. The proposed features uniquely attribute to the specified classes of devices. The Internet traffic classification has gathered momentum from the employment of machine learning. However, with the advent of IoT devices, Internet traffic is going gigantic and heterogeneous. The sophistication of traffic has made the machine learning algorithms inefficient for the IoT-generated traffic. The focus is on the next-generation specialized machine learning algorithms, deep learning. The intelligent learning algorithms can make decision-making similar to the human brain and hold out against the novel traffic flow.

Figure 2 presents the traffic classification roadmap based on the abstract concepts from inception to date.

The accuracy of the solution has always been a prime objective of the classification framework. Nevertheless, in real-time, the following parameters (As depicted in Fig. 3) are vital attributes of the cutting-edge frameworks [12,36]:

1. *On-demand Scalability:* The device classification frameworks should be capable of scaling up or down to address heterogeneous network traffic generated by IoT devices, thereby offering on-demand resources.
2. *Robustness and Portability:* Framework portability refers to operating at various deployment locations, and robustness is associated with consistent prediction accuracy with the emergence of new traffic.
3. *Memory and computational Resources:* The emerging machine and deep learning algorithms require high-end computational resources (e.g., GPU accelerators). Distributed computation is the effort in managing high volume traffic in real-time or short quantum batch jobs.

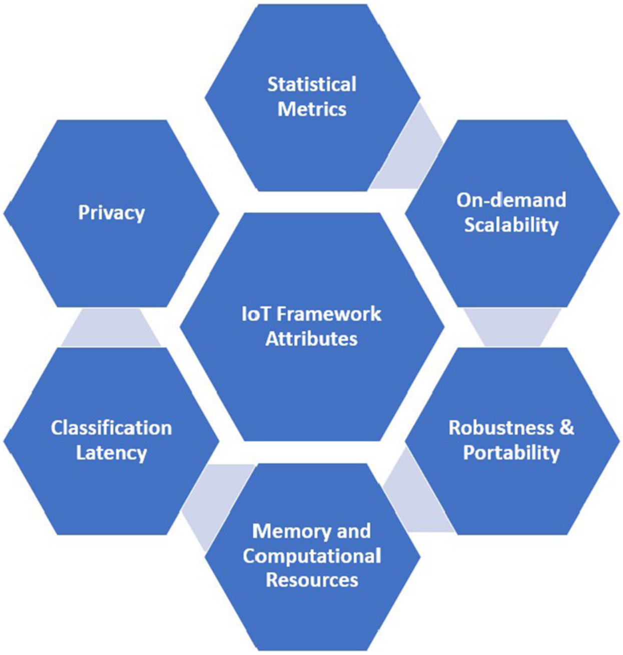


Fig. 3 IoT Classification Framework Attributes

4. *Classification Latency:* The device identification process must be like blazes to offer Quality of Service (QoS).
5. *Privacy:* The privacy policies of an enterprise restrain the information shared on the network. So, the classifier should be able to draw the maximum of non-proprietary traffic attributes.

The traditional machine learning algorithms have proven inefficient with shallow learning algorithms in addressing the aforesaid vital attributes of framework evaluation. The hot off the press development is propelled by the "Big Data" wave supporting streamlined analysis of enormous datasets [23, 27]. Only limited researchers have managed the additional evaluation metrics, and work is in its inception.

This paper has taken the aforesaid vital parameters for the device classification framework into account and developed a distributed, robust, scalable framework based on stack-ensemble methods. The scaling of the computing resources is carried through distributed clusters of Docker containers equipped with H2O prediction models [15] in the AI/ML pipeline. The framework can handle latency and privacy issues. We have further recommended using the proposed framework in global intrusion detections [47], including signature-based attacks [46], anomaly detection [52], and application areas in smart environments.

1.3 Acronyms

The acronyms used in the literature are defined in Table 1:

Table 1 Acronyms used in the literature

Acronym	Description
ANN	Artificial Neural Networks
CAGR	Compound Annual Growth Rate
CNN	Convolutional Neural Networks
DL	Deep Learning
DNS	Domain Name System
DRF	Distributed Random Forest
GBM	Gradient Boost Machine
GLM	Generalized Linear Model
GPU	Graphics Processing Unit
HTTP	Hyper Text Transfer Protocol
HTTPS	HTTP over SSL
IANA	Internet Assign Number Authority
IIoT	Industrial Internet of Things
IoT	Internet of Things
IP	Internet Protocol
ISP	Internet Service Provider
LI	Lawful Interception
M2M	Machine to machine
ML	Machine Learning
MOJO	Model Object
NTP	Network Time Protocol
QoS	Quality of Service
RNN	Recurrent Neural Networks
SMTP	Simple Mail Transfer Protocol
SSL	Secure Socket Layer
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

1.4 Research and development Questions

The locus of research and framework development lies close to the following questions (described in Table 2):

1.5 Our Contributions

We have proposed a novel framework for IoT devices identification and classification. We have brought the novelty forward by proposing the two-stage classifier that integrates best-of-the-class machine learning algorithms. The novel two-stage classifier fits into distributed big data processing engine, which is scalable, portable, and extensible. Furthermore, the highly durable, reliable, and high-performance streaming platform makes the solution highly resilient. The Stack-Ensembled model is the heart of the framework, with computational intelligence distributed to the H2O cluster. More concretely, the paper answers the research and develop-

Table 2 Research and Development Requirements and Responses

Requirement No.	Requirement	Response
RQ1	Can we accurately distinguish the IoT devices from non-IoT devices solely based on behavioral traits of network traffic?	We have proposed the IoT device identification and classification framework entirely based on device activity and signaling behavior. Sections 3 and 4 explains how the behavioral attributes fit into the classification picture.
RQ2	Can the behavioral traffic attributes be applied to distinguish between IoT devices?	Yes. The details for the same are in line with the RQ1 response
RQ3	Is it possible to propose the classification solution as an extensible, distributed, scalable, and portable framework with the growing network traffic?	The paper proposes a distributed device classification framework architecture for computational load distribution. The defense solution is scalable and leverages the scalable Docker containers. The model is implemented as a platform-independent MOJO object and can be easily ported to any H2O-enabled server. The proposed solution can process a high volume of real-time traffic. Section 4 illustrates the architectural details of the framework
RQ4	What effect does excluding payload attributes (to address the privacy issues) have on the solution's performance?	The model behaves well with IoT traffic behavioral attributes only. We have achieved 99.94% accuracy using our proposed novel stack ensemble model. The framework reduces the computational load for analyzing and extracting the packet payload and addresses privacy issues. The performance details are discussed in Sect. 4

ment questions (as described in Sect. 1.4) and has following contributions to the research community:

1. The novel framework includes the stack of state-of-the-art machine learning algorithms ensembled to offer high performance. The paper evaluated several classification models as base models for the Stack Ensemble Metamodel and selected four best-performing models as candidates for the base models. The framework architecture distributes the computational effort to the cluster of intelligent H2O nodes for low latency and high-performance computations.
2. To the best of our comprehension, we are the first to propose a big-data based extensible, distributed, scalable, and portable traffic classification framework. The framework can readily be appreciated as a foundation stone for the security frameworks.
3. The paper contributes a novel architecture to process ingress IoT flow stream. The paper has proposed distributed solution leveraging H2O.ai to act on a network stream. The inclusion of Apache Kafka [5] and integration of scalable Docker-based distributed computing engine (H2O.ai) ensures that the ingress traffic flow is processed in real-time (or near to real-time in batch processing with short quantum). The framework architecture addresses on-demand scalability, memory resources, computational distribution, latency, and privacy requirements, vital for a real-time processing engine. Moreover, the framework components have carefully been selected to be compatible with Big-data tool(s) like Apache Spark for higher performance.
4. The paper comprehended IoT traffic traces' flow level statistical attributes and excluded the payload attributes to avoid privacy infringement.

1.6 Paper Organization

The organization of the rest of the paper is as follows: Sect. 2 explains relevant previous work in a similar area. Sect. 3 presents the intent of the system development and how it fits into the big picture. It also includes the details of the dataset, IoT characterization based on the IoT traffic behavioural study and introduces the novel Stack Ensemble Model. Section 4 presents the experimentation details, discusses the model development and deployment, and concludes the experiment results. Section 5 offers the recommendation and future directions. Finally, Sect. 6 concludes the study.

2 Related Work

Researchers have conducted diligent research towards categorizing Internet traffic for the last two decades. The nucleus of the researcher was the application detection (For example, web applications, Peer-to-Peer, DNS, SMTP, Gaming, etc.) [28,34]. However, the work towards IoT device identification and classification is in inception.

Cirillo et al. [10], in their paper, have classified the IoT traffic flow using a supervised ensemble learning algorithm. They have applied spectral analysis to packet lengths for feature reduction. The authors have used Random Forest Classifier as the base algorithm for ensemble learning and adopted the tenfold cross-validation to evaluate the model performance. Their experimentation results show a high accuracy (98.0%). The high Recall (99.2%) and Precision (99.2%) values underscore the high reliability of the system. Lately, Khadse *et al.* [21] have also used an ensemble learning approach to classify multi-class IoT data collected from diverse domains such as electric grid, air quality sensors, cardiotocography sensors, etc. They have attained an average accuracy of 91.7% for binary classification and 90.6% for multi-class categories. Along the lines, Junior et al. [32] have applied a combination of Large Margin Nearest Neighbour (LMNN) and SVM in the dimensionality reduction process and achieved an accuracy of 95%.

Sivanathan et al. have classified the IoT devices based on the active TCP port scan and proposed a hierarchical TCP scan model for the IoT device classification. In the recent past, the authors have used destination port as a critical attribute for device classification in the extended work [43,45]. They instrumented the lab with 28 IoT devices, such as cameras, sensors, lights, and smart plugs. They have published the traffic traces collected for six months period for the research community [42]. They further characterized the IoT traffic and classified the IoT devices using the rich feature sets, including port numbers, communication protocol, flow volume, activity cycles, etc. The authors have deployed a hierarchical two-stage classifier in the machine learning pipeline, the Naive Bayes classifier for stage-1 and Random Forest Classifier for stage-2, and attained an exceptional accuracy of 99.88%. The authors have further explored their work to detect the behavioral change to IoT devices using the unsupervised clustering technique. In a like manner, to classify the IoT device based on the device behavior, Hamza et al. [17] proposed and developed a tool to generate the Manufacturer Usage Description (MUD) profiles [25] for the IoT devices. Hsu et al. [19] further experimented on the unvarying dataset to identify the class of IoT devices rather than a specific device. The concept is motivated by the fact that the category devices manifest similar traffic behavior. For instance, all motion sensors may show similar traffic behavior. However, the authors did not discuss the details of the performance metrics.

Desai et al. [13] designed a feature ranking classification framework ranking the features based on the cost of dataset retrieval, feature extraction, storage, computational resources, etc. They demonstrated 95% accuracy with eight features and 99% accuracy with 32 markers using the random forest technique. Following the Random Forest, Santos et al. [39] have achieved an accuracy of 99.0% for IoT device classification. Howbeit, they have employed the payload inspection for feature extraction.

Khandait et al. [22] devised a deep packet inspection-based framework for IoT device classification. The framework extracts the unique keywords, such as domain and device names, frequency of occurrence from the device flows. The work reported in [31] is one of the primary studies using machine learning-based classifier. They did experimentation using several classifiers like Random Forest, Gradient Boost (GBM). On a similar trend, [2] used protocol headers to extract the feature-set to apply decision tables for IoT device classification. However, the experimental study infers that several manufacturers implement similar keywords in more than one IoT device type, compromising the solution performance.

Ammar et al. [4] have demonstrated exciting work in their endeavors to classify the near-real-time heterogeneous IoT traffic. They extracted the traffic features from the flow and payload information, applied supervised learning, and achieved an accuracy of 97.0% for autonomous device identification. Marchal et al. [29] have devised a framework based on periodic traffic communication to develop an autonomous classification solution. The authors have shown an accuracy of 99.2% for 33 device types. Their novel technique utilized the periodic IoT traffic communication in the passive fingerprints. The solution has a distributed implementation for high performance. Bao et al. [8] have also invested significant efforts to develop an autonomous solution and identify heterogeneous traffic from IoT devices. They have adopted deep neural networks for model implementation and autoencoders for feature reduction.

There are various methods employed for the classification and prediction tasks. However, the selection of an appropriate model for prediction-related problems is still a challenge.

Izonin et al. [20] employed a novel non-iterative GRNN ensemble model as a prediction method. The new prediction model works on processing the individual dataset by the ensemble members. The meta-algorithm used the SGTM neural-like structure. The experimental results of predicting the used cars prices tasks show that the proposed method is best suited for achieving high accuracy with minimal time and resource costs. Tkachenko et al. [50] continued their work by proposing a GRNN-based prediction approach for missing data recovery jobs. The authors applied the approach to the air pollution monitoring dataset with missing sensor data. They further elucidated the selection of optimal parameters of developed methods and obtained encouraging results with high accuracy.

The aforesaid discussed work makes remarkable contributions, but the prolific traffic volume, distribution of computational loads, real-time processing, framework scalability, and solution portability remained unexplored. Moreover, only a few researchers have followed the discussion around a range of performance parameters. Our work overcomes the shortcomings mentioned above. Table 3 depicts the comparative analysis of the work accomplished to date and our proposed framework. In the table, the symbol ● represents the presence of an attribute, and ○ depicts the absence of the same.

3 System Overview

The section outlines the overview of IoT device identification and classification system. It also describes the components used in each of the stages of the end-to-end ecosystem, such as details of the dataset used for the experimentation study, listing and description of statistical traffic attributes and elements of the stack ensemble model. The ingredients are unfolded as follows:

3.1 Background and the Dataset

The fuel to our implementation is based on several flows codified into packet streams encoded in .pcap (packet capture) format files. The files contain network packets annotated with precise timestamps defining the transmission timestamp and the packet length. The packet capture file is a binary file that stores the packet information as a sequence of bytes, including the protocols headers and the payload information. Our primary source of data is the subset of the IoT trace dataset, released publicly by the work mentioned earlier by Sivanathan et al. [43,45]. The IoT traces contain six months of network trace data from 21 IoT and seven non-IoT devices instrumented in the university lab environment. Each of the packet-capture files that the authors release corresponds to one day of flow information. We refer to the dataset as the ***"Sydney UNSW dataset."*** [1]. The subset of the devices obtained from the packet capture file used for model development and prediction are shown in Table 4.

The .pcap file that we have used for our experimentation purpose contains 17,51,968 packets in binary format. In order to probe the traffic features from the IoT traces, we have converted the row packet captures (.pcap) into flow entries using the Cisco Joy tool [14]. We further developed a python script to process the flow entries and compute the statistical features for IoT devices. The statistical attributes contain the device's behavioral activity patterns and signaling attributes. We have initially extracted the 23,987 instances of flow entries from the IoT devices. After pre-processing the flow entries, we rolled 21,598 flow entries from 13 IoT devices and six non-IoT devices (labeled non-IoT) for algorithm training, testing, and validation purposes. Figure 4 shows the percentage distribution of the underlying protocols in the device flow. We have validated the solution using another .pcap file containing 8,02,582 packets from 17 devices in binary format.

3.2 IoT Traffic Characterization

The section presents the details of the behavioral aspects of the IoT traffic based on the passive packet analysis. We have studied the Sydney UNSW IoT traces dataset that includes traffic from 28 devices over six months for the IoT traffic characterization. The traffic behavior is categorized based on the following attributes:

3.2.1 Statistical Attributes

The statistical analysis of IoT traffic relies on the network traffic attributes such as the number of bytes flowing in or out of the interface, packet sizes, inter-arrival time for packets, etc. Several researchers use these statistical attributes to derive traffic patterns, and more meaningful features [2]. The technique does not require looking at the payload contents. It is computationally fast and adopted by many machine learning algorithms. The advantage of the method is that we can deduce the class of unknown devices. The attributes are further categorized into:

1. *Attributes based on data exchange:* We examined that the critical traits of IoT traffic at the per-flow level are Flow Volume and Flow Rate as defined below:

$$Total_{FlowVol} = B_{In} + B_{Out} \tag{1}$$

Where,

$Total_{FlowVol}$ is total number of bytes exchanged during a flow.

B_{In} is total number of bytes flowing in the network interface per device per flow.

B_{Out} is total number of bytes flowing out of the network interface per device per flow. And,

$$Rate_{Flow} = \frac{Total_{FlowVol}}{Dur_{Flow}} \tag{2}$$

Where,

Dur_{Flow} is duration of the flow.

[1] The dataset is accessible at the following URL: https://iotanalytics.unsw.edu.au/iottraces.

Table 3 Comparative Analysis of existing solutions and our proposed framework

Reference(s)	Performance Metric(s)						Scalability	Portability
	Accuracy	Precision	Recall	F1-score	Other P. Metrics	Performance comparison		
[45]	●	○	○	○	○	○	○	○
[10]	●	●	●	●	●	●	●	○
[21]	●	○	○	●	○	○	○	○
[44]	●	○	○	○	○	○	○	○
[43]	●	○	○	○	●	○	○	○
[19]	●	○	○	○	○	○	○	○
[13]	●	○	○	●	●	○	○	○
[39]	●	●	●	●	●	●	○	○
[22]	●	○	●	○	●	●	○	○
[31]	●	○	○	○	●	○	○	○
[2]	●	○	○	○	●	○	○	○
[4]	●	●	●	○	●	○	○	○
[29]	●	●	●	●	●	○	○	○
[8]	●	○	○	○	●	○	○	○
Our Proposed Framework	●	●	●	●	●	●	●	●

Table 4 IoT and Non-IoT device list and MAC mapping

IoT Device	MAC Address	Non-IoT Device	MAC Address
Amazon Echo	44:65:0d:56:cc:d3	Samsung Galaxy Tab	08:21:ef:3b:fc:e3
Smart Things	d0:52:a8:00:67:5e	MacBook	ac:bc:32:d4:6f:2f
Netatmo Camera	70:ee:50:18:34:43	Laptop	74:2f:68:81:69:42
Samsung Smart Cam	00:16:6c:ab:6b:88	Android Phones	40:f3:08:ff:1e:da
Dropcam	30:8c:fb:2f:e4:b2	iPhone	d0:a6:37:df:a1:e1
Insteon Camera	00:62:6e:51:27:2e	MacBook/iPhone	f4:5c:89:93:cc:85
Belkin Wemo Switch	ec:1a:59:79:f4:89		
Belkin Wemo Motion Sensor	ec:1a:59:83:28:11		
NEST Protect Smoke Alarm	18:b4:30:25:be:e4		
Netatmo Weather Station	70:ee:50:03:b8:ac		
Withings Smart scale	00:24:e4:1b:6f:96		
Withings Aura Smart Sleep Sensor	00:24:e4:20:28:c6		
Light Bulbs LiFX Smart Bulb	d0:73:d5:01:83:08		
Triby Speaker	18:b7:9e:02:20:44		
PIX-STAR Photo-frame	e0:76:d0:33:bb:85		
HP Printer	70:5a:0f:e4:9b:c0		

To study the device behavior based on data exchange, we have plotted the flow volume (As shown in Fig. 5a) exhibited by Amazon Echo, Belkin Switch, and Insteon Camera. However, we have verified that other IoT device categories also possess similar behavior. As we observed from the flow volume plot, for Amazon Echo, more than 87 percent of data transfer is less than 100 bytes per flow. Similarly, Belkin Switch has a data transfer between [200-650] bytes for 95% of the flows, and Insteon Camera reported the data transfer between [100–300] bytes per flow for more than 83% of the flow.

Likewise, we observed a similar pattern for the average flow volume. Quantitatively, Fig. 5b shows that Amazon Echo maintains the flow rate of fewer than 160 bytes, whereas, for Insteon Camera, the flow rate is less than 500 bytes with the lower limit of 30 bytes. However, the Belkin switch maintained a higher flow rate than Amazon Echo and Insteon Camera.

2. *Attributes based on device activity:* Other critical statistical features that are based on device activity are Flow Duration and Device Sleep Time.

$$Dur_{Flow} = F_i^{End} - F_i^{Start} \tag{3}$$

Where,

Dur_{Flow} is the total duration for which flow happened.

F_i^{End} is flow end time for i*th* flow entry.

F_i^{Start} is flow start time for i*th* flow entry.

And,

$$T_{Sleep} = F_{i+1}^{Start} - F_i^{End} \tag{4}$$

Where,

T_{Sleep} is the time duration for which the device was not active and is considered the time between two consecutive flows.

F_{i+1}^{Start} is start time for (i + 1)*th* flow entry.

F_i^{End} is end time for i*th* flow entry.

The IoT devices send data in small bursts and sleep for a specific duration. We observed that flow duration for more than 80% of the flows lasted less than a second. The fact that non-IoT devices have a higher flow duration makes the flow duration a critical parameter. Figure 5c shows that the devices considered for illustration exhibit similar behavior for the entire flow period. Lastly, the sleep pattern in Fig. 5d ascertains that the categories of IoT devices show identical sleep patterns. Amazon Echo reveals relatively short device sleep times because it keeps TCP connections alive and sleeps only in case of disconnection.

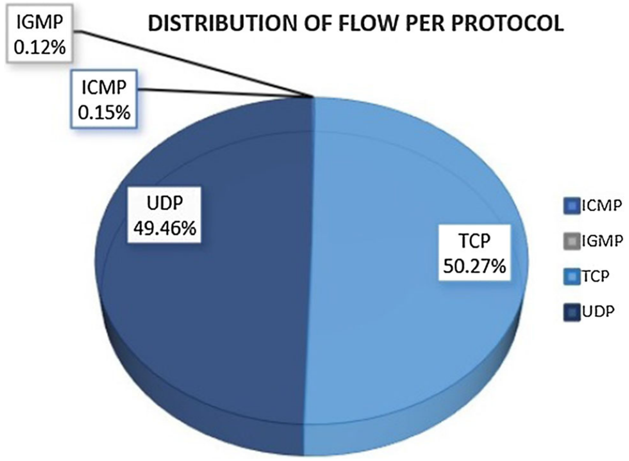


Fig. 4 Percentage distribution of flow entries per protocol

3.2.2 Classical Port-Scanning

The classical port scanning technique is fast, requires low computational resources, and is supported by most networked devices. The port numbers do not implement a payload at the application layer. Hence, user privacy is not compromised. The IoT devices are special-purpose, low-power devices accessing limited remote services and port numbers. The port numbers reserve special attention in IoT device identification and characterization [44].

The study from the dataset under investigation verifies the importance of the destination port number in the feature list. Figure 6 shows the pictorial presentation of port numbers and the number of IoT devices accessing them.

Furthermore, it also infers from the flow that port numbers 80, 123, 443, and 49153 are the most widely used. We took Amazon Echo, Belkin Switch, and Insteon Cam for illustration purposes in Fig. 7 and presented the frequency of access of each port number. The instances show that the most requested ports numbers are HTTP (port 80) and HTTPS (443) ports, whereas the devices use port 123 (used for NTP protocol) to query and sync with the time-server quite frequently.

We also note that devices from the same manufacturers used a standard set of port numbers. For example, Belkin

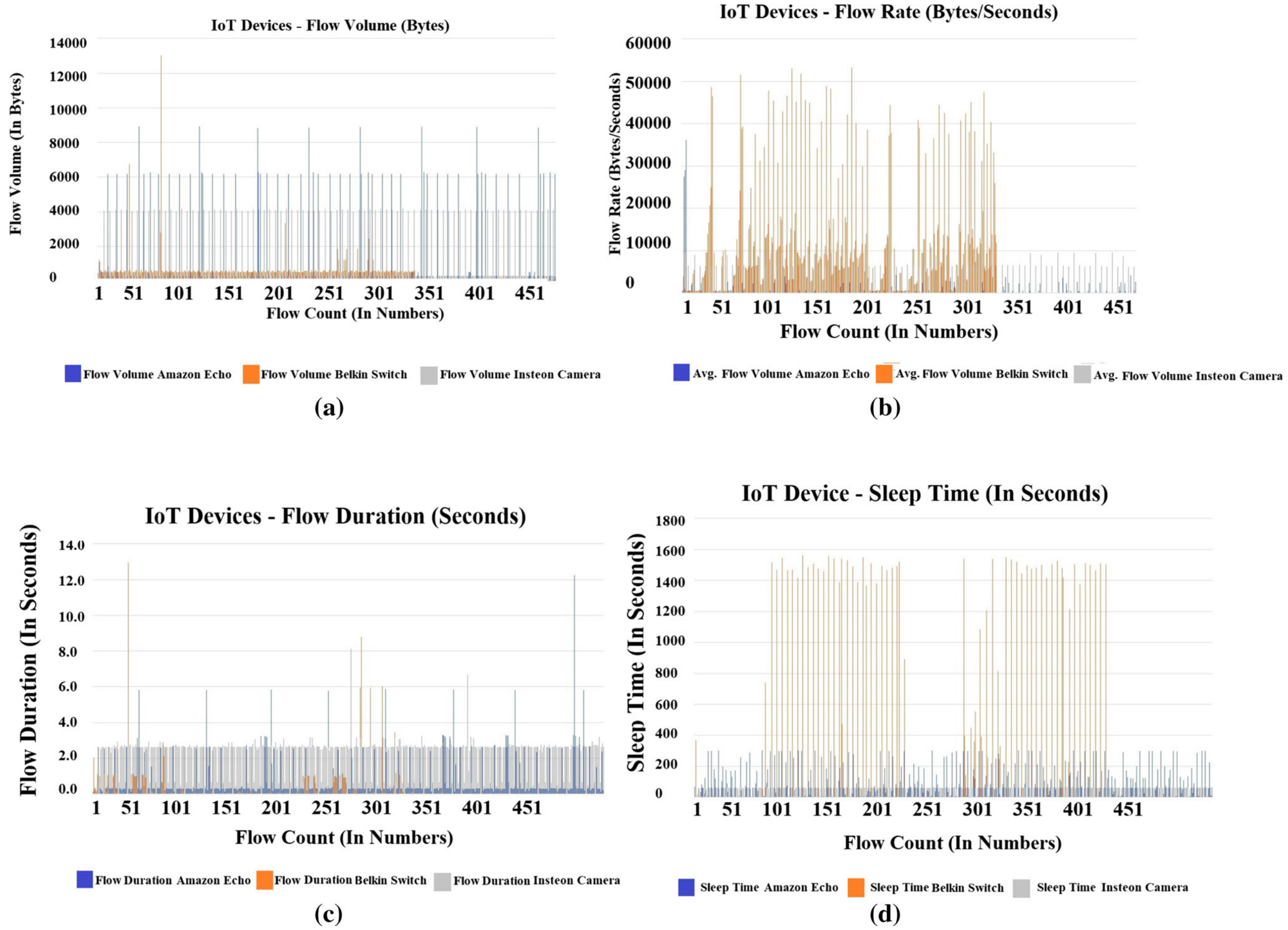


Fig. 5 Statistical flow attributes **a** Flow Volume, **b** Flow Rate, **c** Flow Duration, **d** Device Sleep Time

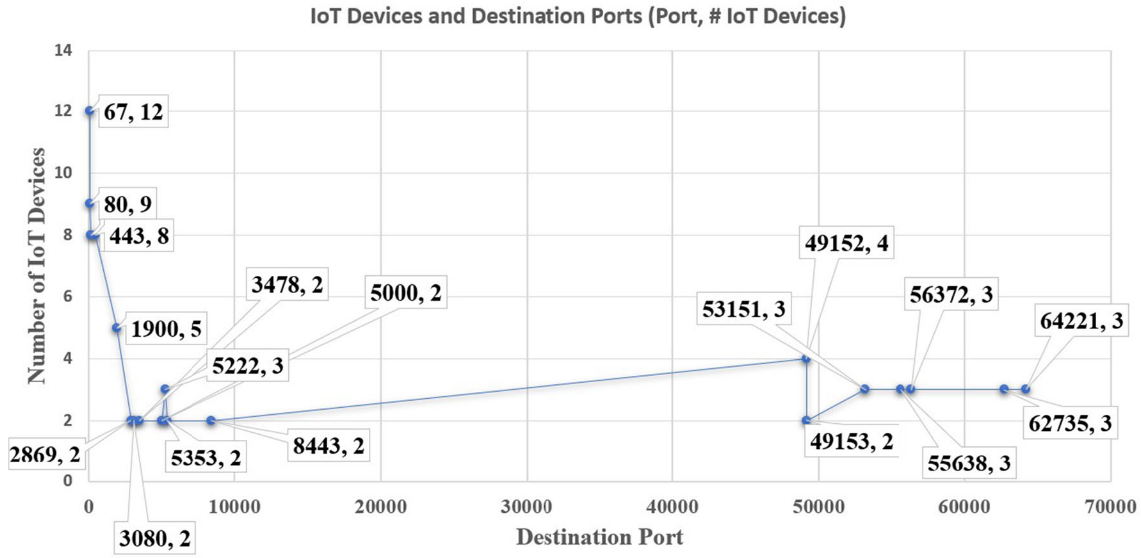


Fig. 6 Destination ports and number of connected devices

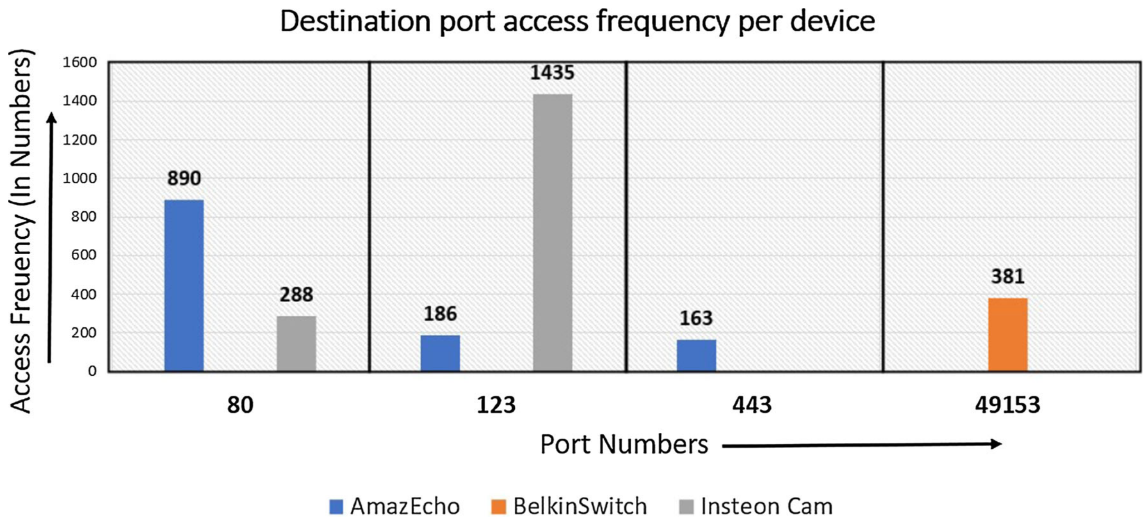


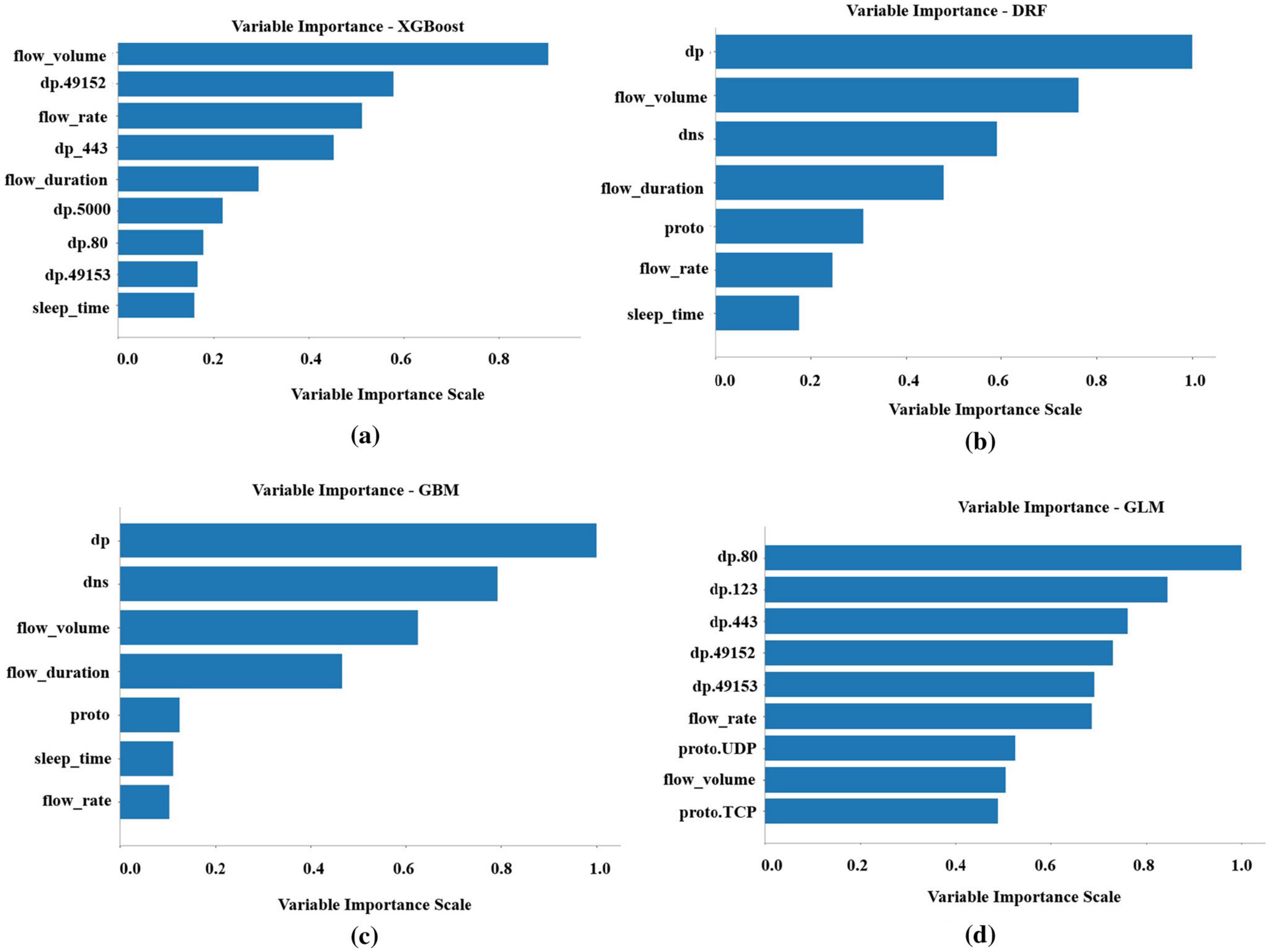
Fig. 7 Destination ports and number of connected devices

Wemo Motion Sensors and Belkin Wemo Switch use port numbers 3478 and 8443. We also note that several IoT devices avail standard network services DNS (53), NTP (123), ICMP (0), and SSDP (1900) on the well-known service ports.

Our experimental test results (In Sect. 4) also show that destination ports have a high significance in intelligent model building. Figure 7 depicts the variable importance graphs in the machine-learning models used in our experimentation setup. The figure contains the labeling of each of the variable importance plots with the corresponding model. More details are available in the experimentation section. We further deduce from the plots that Gradient Boosting Machines (GBM) algorithm Fig. 8c has placed destination ports on the high ranking in the variable importance plot. XGBoost Fig. 8a and General Linear Model (GLM) Fig. 8d have emphasized individual ports as individual features in the feature list, with ports 49152, 80, 123, and 443 placed at the higher positions on similar lines. Distributed Random Forest Fig. 8b upholds the port number importance as in the GBM model.

3.2.3 IoT Functional Attributes:

The functional attributes are the leitmotifs to perform device management activities. IoT devices are special-purpose devices designed for a specific objective. Contrary to non-IoT devices, they access limited domains corresponding to the service provider realm. We observed from the dataset study that each device accesses a circumscribed sphere of URLs that are generally unique to the device. For example, Amazon Echo accesses *example.org, example.com*, and



(a)

(b)

(c)

(d)

Fig. 8 Variable importance plot

example.net as the most communicated domains. Moreover, the device is observed accessing the mentioned domains over a frequency of 5 minutes. Likewise, *pool.ntp.org* is the frequently accessed domain (for NTP queries) for Smart Things devices. Conversely, non-IoT devices access diverse and non-unique domain names.

Another critical operative attribute of IoT devices is managing accurate and reliable time. The majority of IoT devices communicate with standard NTP servers over UDP protocol (port 123) [33] to synchronize time in a periodic manner [53]. More of the periodic and repetitive operative tasks involve negotiating for SSL connection, exchanging *KeepAlive* messages, etc.

3.3 Stack Ensembe Model

This section explains our proposed novel Stack Ensemble algorithm to classify IoT devices based on behavioral traffic features. The Stack Ensemble paradigm works with the intent of combining several models. The models are combined to reduce the bias and variance. The base models are stacked in a two-layer architecture as depicted in Fig. 9. Stage-0 represents the base model and data, whereas stage-1 refers to the meta-model and the cross-validated data. We have used diverse base models to get the maximum out of stacking. The model employs k-fold validation to parameter fine-tuning and evade the information leakage. We intend that each base model train (k minus 1) folds and apply the prediction on the untrained fold.

The input dataset is flow entries from the ingress network traffic. Let D be the flow entries where:

$$D = \{(X_1, Y_1), (X_2, Y_2), (X_3, Y_3), ..., (X_N, Y_N)\}$$

In the input flow entries,

$$X_i \in F\,(Feature\,set) = \{f_1, f_2, f_3, ..., f_n\},\ and$$
$$Y_i \in P\,(Predicted\ Device\ Labels) = \{P_1, P_2, P_3, ..., P_n\}$$

We have trained the Stack Ensemble model with the learning algorithm set:

$$L_i = \{ L_1, L_2, L_3, ..., L_n \}$$

The learning algorithms set values as defined as follows:

$$L_1 = \{ XGBoost_1, XGBoost_2, XGBoost_3 \}$$

i.e., XGBoost is trained in three of the pre-specified configurations.

Similarly,

$$\begin{aligned} L_2 &= \{ DRF \} \\ L_3 &= \{ GBM_1, GBM_2, GBM_3, GBM_4, GBM_5 \} \\ L_4 &= \{ GLM \} and \\ M_{MetaModel} &= \{ StackEnsemble\_MetaModel \} \end{aligned}$$

Algorithm 1 describes the algorithm details as follows:

```
Algorithm 1: ClassifyIoTDevices
Input:
    $D = \{ (X_1, Y_1), (X_2, Y_2), (X_3, Y_3), ..., (X_N, Y_N) \} \quad \rightarrow Flow\ Entries$
    $L_i = \{ L_1, L_2, ..., L_N \} \quad \rightarrow Layer1\ Models\ (\ Base\ Algorithms\ )$
    $M \quad \rightarrow Meta\ Learning\ Algorithm \quad \rightarrow Layer2\ Model\ (\ Meta\ Algorithms\ )$
    $N_{Base_Models} = Number\ of\ Base\ Models$
    $N^{Folds} = Number\ of\ folds$
Output:
    $Trained\ Base\ Models\ L_i$
    $Trained\ Meta\ Model\ M$
Split D into $X^{train}, X^{test}$ → (Training and Testing Set)
$X^{Meta} = An\ empty\ list\ of\ size\ N_{sample} X N_{Base_Models}$
while $m$ in $L_i$ do
    while count in $K_{folds}$ do
        Train model $m$ on $(k-1)$ folds
        Predict on k$th$ fold
        Add predictions to $X^{meta}$
    end
end
Train M (Meta Model) on $X^{meta}$
return Trained $(L_i, M)$
```

4 Experimental Details and Results

We have targeted the system model (As shown in Fig. 10) of a typical IoT network connected to an IoT cloud service. The primary goal of the system model is to identify the type of connected devices so that system administrators can ensure the effective operations and enforcement of policies to secure the network. The core of device identification relies on passive monitoring of the network traffic generated by IoT and non-IoT devices. The framework does not use active or dynamic probing of the devices. The active probing techniques are generally employed and tailored for a specific class of IoT devices that require special permissions from the device owners. The model meets the requirements of autonomy, stability, low classification latency, scalability, and portability.

4.1 Experimental Testbed

In this section, we present the experimental testbed. We instrumented the testbed to design, train, and deploy the classification model. The testbed consists of virtual and physical hosts, distributed H2O.ai containers. We created the virtual hosts using Oracle VM VirtualBox [1]. We have employed 04 virtual hosts in the testbed. All the hosts are Ubuntu 20.04 machines with 4 GB RAM, Quad-core i5 1.60 GHz processors. The details of the individual host (As presented in Fig. 11) and their role is as below:

- *01 Host:* For Model Training and Model Development. We have further utilized the same host for feeding the flow stream from the dataset to Apache Kafka Server. As

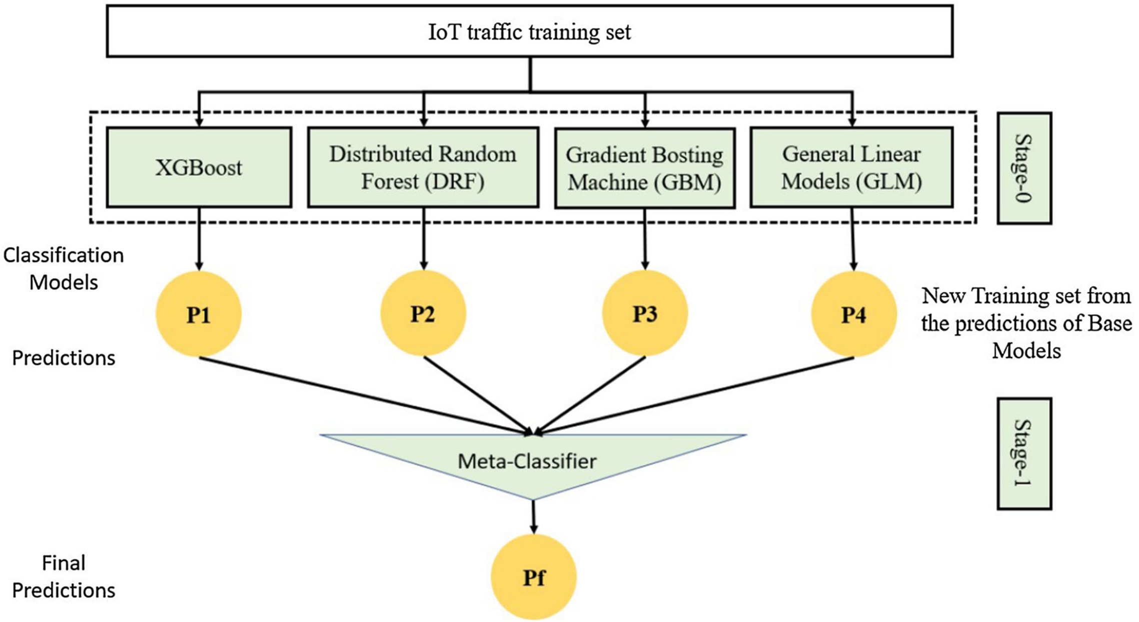

IoT traffic training set
XGBoost
Distributed Random Forest (DRF)
Gradient Bosting Machine (GBM)
General Linear Models (GLM)
Stage-0
Classification Models
P1
P2
P3
P4
New Training set from the predictions of Base Models
Predictions
Meta-Classifier
Stage-1
Final Predictions
Pf

Fig. 9 Stack Ensemble architecture for IoT device classification

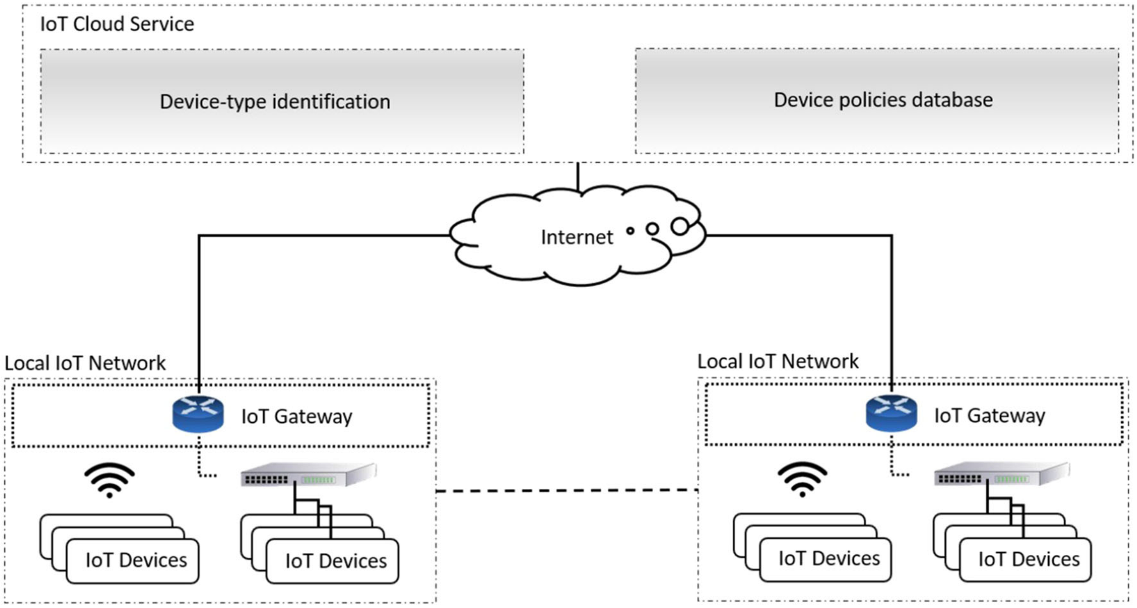


Fig. 10 IoT Device Classification System Model

explained earlier, we have referred to the packet capture dataset from the real-life smart-environment setup.

- *01 Host:* The machine hosts the Apache Kafka server for streaming. The Kafka host is responsible for running the Zookeeper and Kafka broker. Kafka hosts the streaming topic, *"iot_device_live_stream"*, for producer and consumer applications to subscribe.
- *01 Hosts:* We created a network for H2O.ai docker containers and added 3 H2O.ai docker containers for distributed H2O.ai cluster implementation.
- *01 Host:* The machine hosts the binary Stack Ensemble model (a.k.a. MOJO The Model Object), receives a stream of Kafka topic flows, and makes predictions.

The flow for the IoT classification pipeline starts with the ingress packet capture (.pcap files). The native Cisco tool converts the packet capture into JSON flow entries and feeds to JSON parser. JSON parser converts the JSON flow entries into CSV formatted flow entries. The flow entries are further pre-processed to handle missing entries, hot encoding,

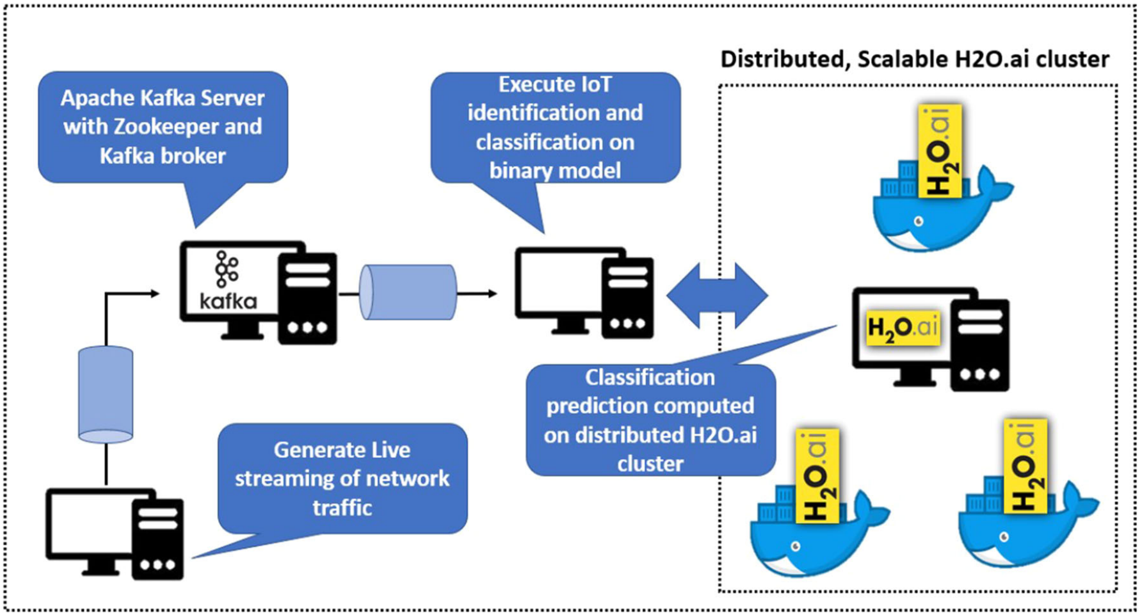

Distributed, Scalable H2O.ai cluster
Apache Kafka Server with Zookeeper and Kafka broker
Execute IoT identification and classification on binary model
kafka
H2O.ai
Classification prediction computed on distributed H2O.ai cluster
Generate Live streaming of network traffic

Fig. 11 Testbed for proposed IoT classification framework

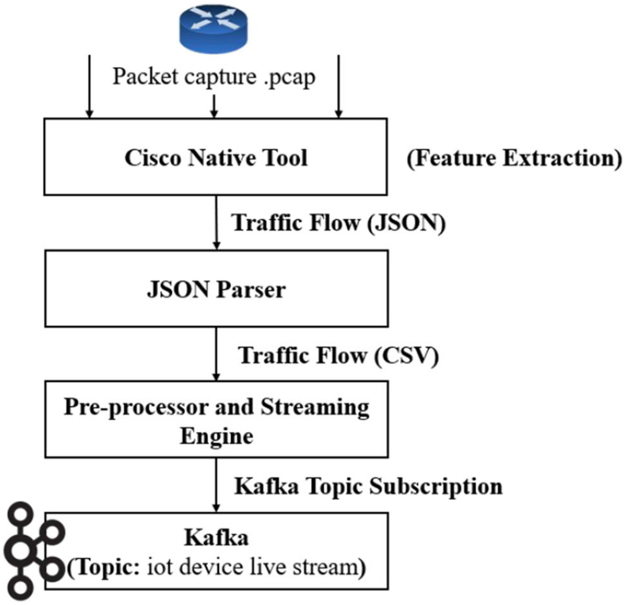


Fig. 12 Setup details for traffic capture and streaming to Kafka engine

NaN (not a number) values, and perform other pre-process operations. The streaming engine produces the flow entries to subscribed Kafka topics (As shown in Fig. 12). The deployed ensembled model consumes the subscribed Kafka topic for the prediction task (As discussed in the later section of the text in Sects. 4.2 and 4.3).

4.2 Model Development

We have constructed the framework in a two-stage process: (a) Model development and (b) Model deployment. The overall process flow of the solution is depicted in Fig. 13.

The details of each stage are as follows: In the proposed solution deployment, we intercept the stream of the live network traffic. However, in the experimentation study, we have used the .pcap file available as a publicly released dataset. In the real-world scenario, on-the-fly network traffic attributes extraction requires the infrastructure to have sufficient visibility into the network traffic flows. In modern-day network infrastructures, such as SDN switches or NetFlow capable devices, we can extract the behavioral attributes (such as flow_volume, flow_duration, etc.) from the flow relatively easily.

For the model development (Column 2 of Fig. 13), we have leveraged the H2O.ai framework for model development and deployment. H2O.ai is an open-source platform, employs distributed computational architecture specifically designed for processing enormous datasets. During the training phase, we have included XGBoost, Distributed Random Forest (DRF), Gradient Boost Machines (GBM), Generalized Linear Model (GLM) as base algorithms. We further divided the IoT trace datasets into a training set (70%), test set (15%), validation set (15%). Thence, we have trained and cross-validated (with nfolds = 5) the models with (a) three specified H2O configurations for XGBoost, (b) GLM, (c) DRF, and (d) five GBM configurations. H2O.ai includes the pre-specified model configurations for quick results for each of the algorithms.

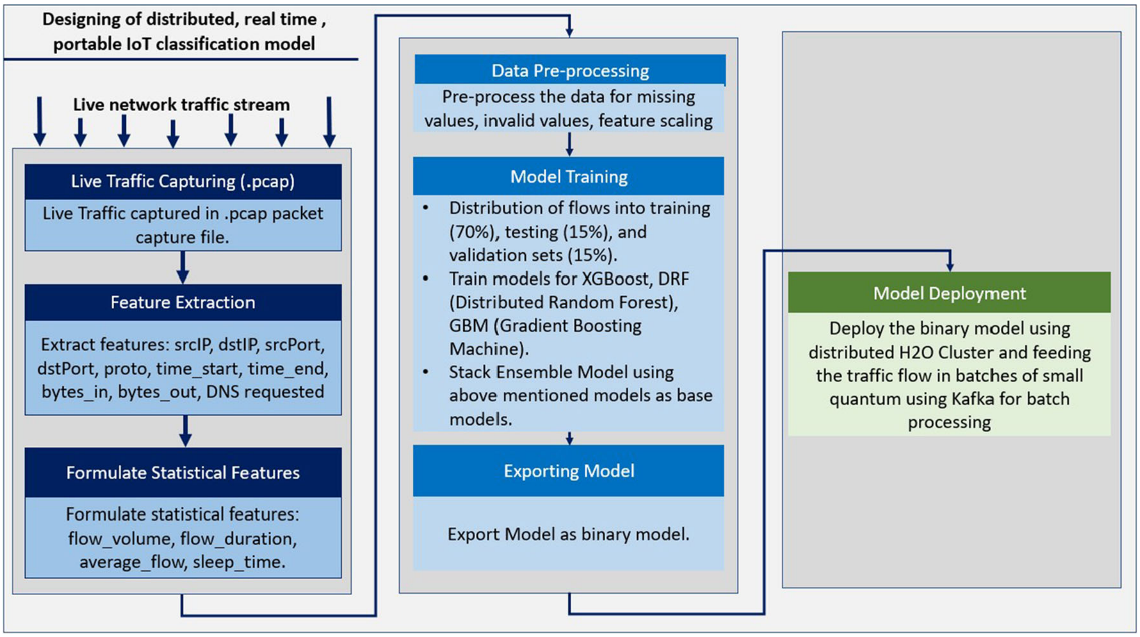

Designing of distributed, real time , portable IoT classification model
Live network traffic stream
Live Traffic Capturing (.pcap)
Live Traffic captured in .pcap packet capture file.
Feature Extraction
Extract features: srcIP, dstIP, srcPort, dstPort, proto, time_start, time_end, bytes_in, bytes_out, DNS requested
Formulate Statistical Features
Formulate statistical features: flow_volume, flow_duration, average_flow, sleep_time.
Data Pre-processing
Pre-process the data for missing values, invalid values, feature scaling
Model Training
• Distribution of flows into training (70%), testing (15%), and validation sets (15%).
• Train models for XGBoost, DRF (Distributed Random Forest), GBM (Gradient Boosting Machine).
• Stack Ensemble Model using above mentioned models as base models.
Exporting Model
Export Model as binary model.
Model Deployment
Deploy the binary model using distributed H2O Cluster and feeding the traffic flow in batches of small quantum using Kafka for batch processing

Fig. 13 Ensemble Learning model process flow

Stack Ensembles involves training a second-level meta-learning process to determine the optimal blend of base algorithms. In our experimentation setup, we trained the meta-learner using fivefold cross-validated base learners. The final version of Stack Ensemble includes the best-performing base models from each class of algorithms. The final version of stacking in our configuration is the instance of the Super Learner algorithm.

4.3 Model Deployment

After building and training the novel Stack Ensembled IoT classification model, we exported it as MOJO (Model Object). We deployed it in the machine-learning pipeline as shown in Fig. 14. We have written a script to subscribe to the Kafka topic and send an ingress stream of network traffic flows to the subscribed topic. Kafka is the prominently used streaming server for reliable, low-latency, high-throughput, real-time network feeds. Kafka server runs on a separate host in the experimental lab. However, it can easily be deployed on a cluster for higher load. We deployed the device identification and prediction module on the production server that lies on the consumer end of Kafka and reads the stream in real-time. The module mentioned above is responsible for loading the exported MOJO object and performing the prediction. The computations are performed on the H2O.ai cluster. To model the H2O server fit into scalable architecture, we deployed the distributed configuration of the H2O server onto the Docker containers, which are highly scalable.

As shown in Fig. 11, the machine-learning pipeline in the production environment starts by consuming the live stream of network traffic, pre-processed by an intermediate module, and produced to Kafka topic. The other end of the pipeline is responsible for munching the live traffic stream and identifying the IoT devices.

4.4 Results and Discussions

We have evaluated the intelligent models over multiple dimensions. The study compares the mean per class error values, RMSE, Logloss, learning curve evaluation, confusion matrix, the model hit ratios, accuracy, precision, recall, F1-score. The multi-dimensional evaluation intends to project the isometric view of the model performance and the experimental results. The model performance dimensions are explained as follows:

4.4.1 The Models Leaderboard

We trained the Stack Ensemble and the base models with the training data during the training and development stage. We prepared and evaluated each of the models for a set of pre-specified configurations. The inputs models were then considered for the best of the family and included as the base model. Table 5 below shows the leaderboard and the evaluation metrics for the input models:

The statistical details in the table show that Stack Ensemble assumes a small Mean Per Class Error value of 0.0183571, which is the least compared to other models.

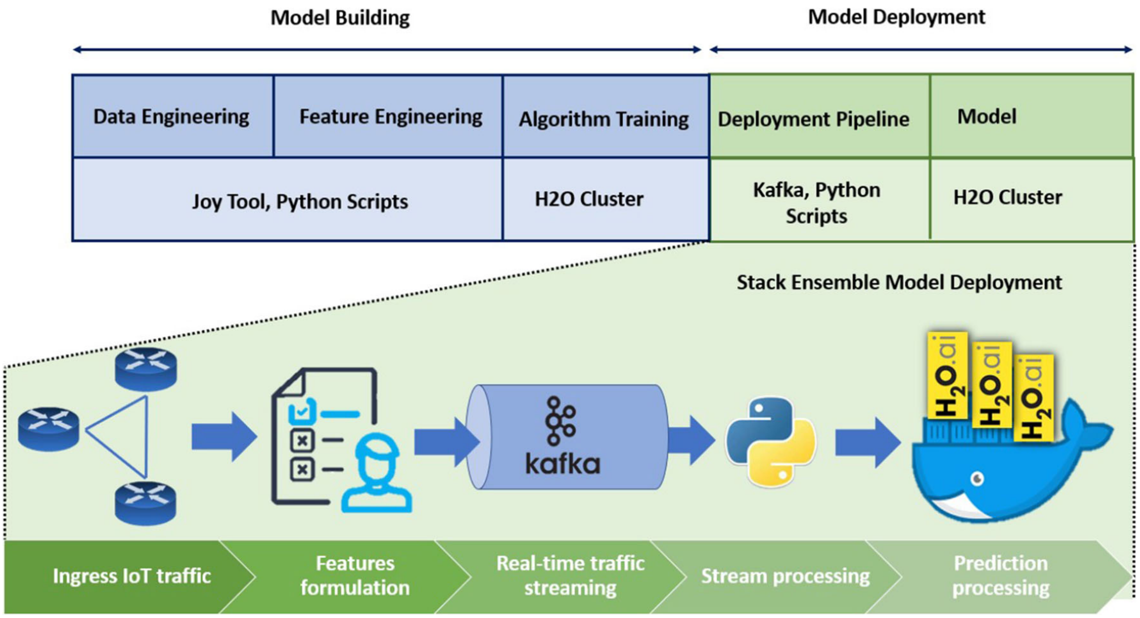


Fig. 14 Novel Stack Ensemble Model deployment

Table 5 Base and Meta Models leaderboard

Model Id	Optimal Parameter(s)	Mean per Class Error	Logloss	RMSE	MSE
StackedEnsemble_AllModels	–	0.0183571	0.0316226	0.0876373	0.00768029
GBM_5	number_of_trees = 96, min_depth = 1, max_depth = 15, min_leaves = 1, max_leaves = 71, mean_leaves = 58	0.0202543	0.0450185	0.096788	0.00936791
XGBoost_3	number_of_trees = 120	0.0224418	0.0373026	0.0942976	0.00889204
DRF_1	number_of_trees = 46, min_depth = 5, max_depth = 20, min_leaves = 6, max_leaves = 274, mean_leaves = 76	0.0313595	0.0743251	0.124194	0.0154243
GLM_1	family = multinomial, $\lambda = 0.002609$, $\lambda_{Max} = 18.99$, $\lambda_{Min} = 0.0026$	–	–	0.37236	0.13865

Hence, each class's small average error value indicates the better Mean Square Error (MSE). The classification predictions on the regression line are closer to the actual values and have the slightest deviations. Likewise, the comparison of the Logloss value of base models and Stack Ensemble also demonstrates that the Meta Model results (Stack Ensemble) are closer to the actual values.

The table also shows all the optimal parameters and the optimal error values for the investigated methods. The parameter list specifies the maximum number of trees and the minimum and the maximum depth of the tree leaves in a parallel distributed environment. The GLM specifies the strength of regularization to avoid overfitting problems.

4.4.2 Model Learning Curves

To study the model in the context of learning performance, Fig. 15 represents the learning efficiency of each of the selected models over some time. The models are evaluated on the training dataset and untrained validation set at the end of each update. The learning curve for XGBoost shows that the training data has a lower deviation of the predicted values at the curve's start. However, a higher variation is observed for the untrained validation set. The curve settles with a lower deviation value. However, the validation and training sets have an observable deviation in the predicted values. We also observed that the learning curve for the GBM model parallels that of XGBoost. The higher values of Logloss in the training set for DRF and the lower Logloss on the validation set reveal that the model performed well on the validation set. Inconsistent with the other models, GLM is not perceived to perform well and has a higher deviation towards the curve's stabilization end. We can see the sustained results with The Meta Model (Stack Ensemble). The model reflects quite a slight deviation towards the curve's tail. The model performs alike for the training, testing, and validation set.

4.4.3 Confusion Matrix

We have further explored the model in the dimension of the primary building block of the performance metrics (True Positives (TP), True Negatives (TN), False Positives (FP), False Negatives (FN)). The vertical axis represents the actual class, and the horizontal axis represents the predicted class. Furthermore, we have encoded the devices per the following dictionary for the readability purpose in the confusion matrix plot:

IoT devices = {0: "Amazon Echo", 1: "Belkin Wemo Motion Sensor", 2: "Belkin Wemo Switch", 3: "Insteon Camera", 4: "NonIoT", 5: "PIXSTAR Photo Frame", 6: "Smart Things", 7: "Withings Aura Sleep Sensor"}.

We have also included the validation dataset comprising the data from an entirely different set of IoT traces to attain the purpose. We plotted the confusion matrix (as shown in Fig. 16) for the test data (Stack Ensemble) and benchmark hold-out of validation data (Base and Meta Models). The confusion matrix upholds the Stack Ensemble's best performance on the test and the validation set. However, the results of GLM are not encouraging and take a higher error rate into account. We observe the optimum performance of Stack Ensemble by the stacking of the base models.

4.4.4 Base Models Hit ratio

Hit ratio is the number of accurate predictions in proportion to the total predictions. The Hit-Ratio for the base models is depicted in Table 6. Though the base models show higher Hit-Ratios for the instrumented IoT devices, all the models interpreted Withings Aura Sleep Sensor correctly in all the prediction attempts. The justification for the perfect hit ratio is that the Sleep Sensor exhibits distinctive flow patterns compared to other IoT devices. The device records a higher flow volume (average 1700 bytes per flow) and an average device sleep time of 5 minutes.

The Hit-Ratio's comparative plot (as shown in Fig. 17) shows that the base models are observed to perform a consistent Hit-Ratio for the instrumented IoT devices. However, the models have shown a variation for the Amazon Echo device. The rationale following the Hit-Ratio deviation is in the way models depicted the flow pattern of Amazon Echo. The device keeps the TCP connections active and sleeps only in disconnection, incorrectly classifying predictions into the non-IoT category.

4.4.5 Essential Performance Metrics

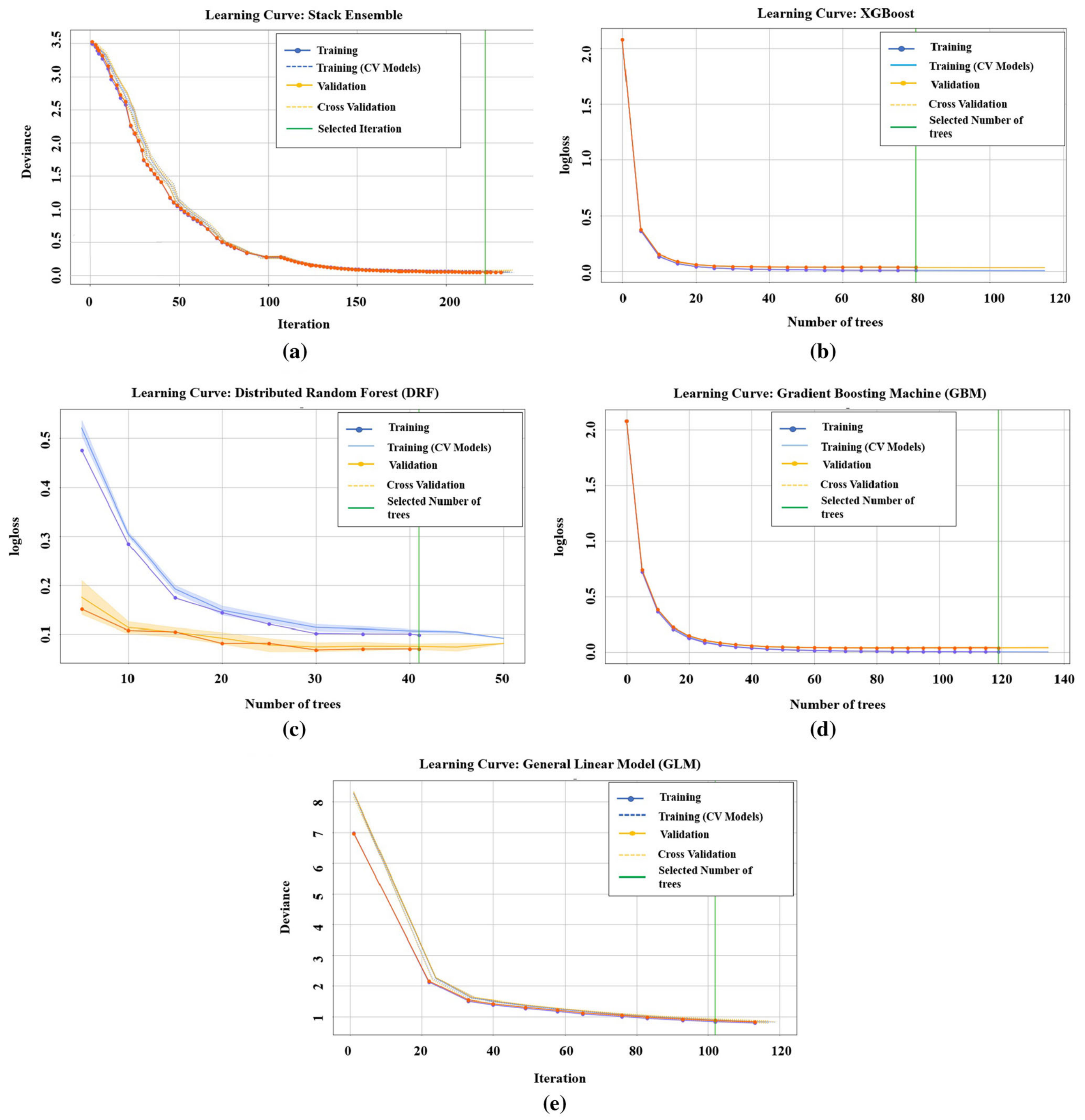
The final dimension in the isometric view of the model performance discussion is the details of the essential performance metrics (Accuracy, Precision, Recall, F1-score). Table 7 shows the Accuracy, Precision, Recall, and F1 score for the Stack Ensemble model for the instrumented IoT devices in the flow entries. We have demonstrated an overall model accuracy of 99.93%. The higher values of Precision, Recall, and F1 score complements the Accuracy of the model. The mentioned table shows that although the Accuracy of the Withings Aura Sleep Sensor is high, the slightly less fraction of Precision infers that certain sleep sensor instances do not belong to the sleep sensor class. The samples misclassified to sleep sensor was from the non-IoT class. We can see the evidence of the model behavior in the F1 score parameter for the Withings Aura Sleep Sensor class.

Table 8 shows the comparative analysis of the base model's Accuracy, Precision, Recall, and F1 score. The GLM is observed to perform poorly as compared to other models. Also, the low precision in the case of GLM is evident that a fraction of devices belonging to a specific class classified by GLM did not belong to that class. The low Recall fraction indicates that the model did not perform well as compared to other models. However, the combined stacking effect of the base model listed the high performance of the Stack Ensemble.

Table 9 shows the comparison of proposed work with state-of-the-art in IoT classification domain.

5 Recommendations and Future Directions

The locus of the research is IoT traffic characterization, device identification, and classification. However, the work paves the opportunities for researchers to unfold the frame-



(a)

(b)

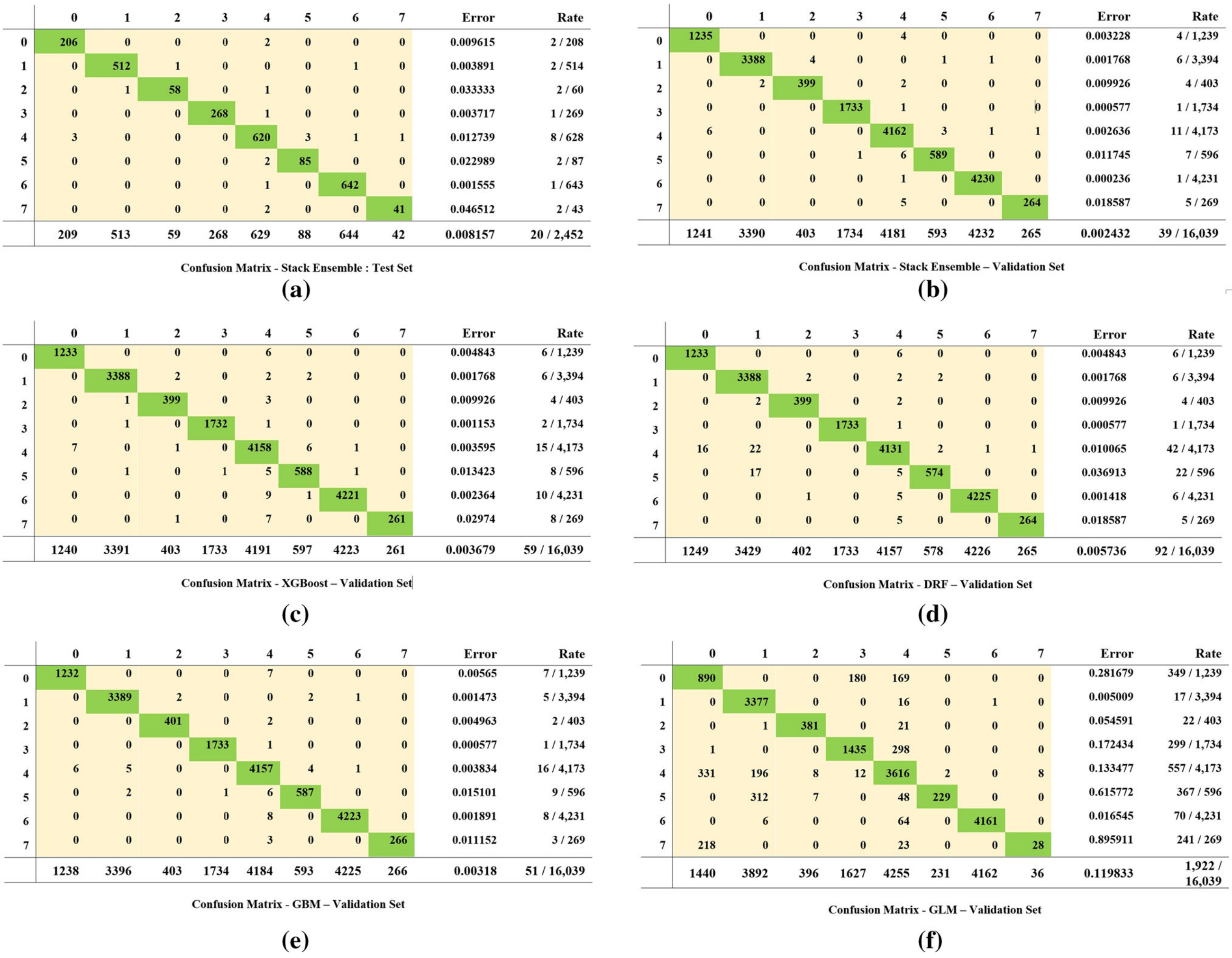
(c)

(d)

(e)

Fig. 15 Learning curve for base and meta models

work architecture and apply it to various problems. IoT devices are omnipresent, exponentially increasing, and geographically distributed. The pervasiveness of IoT devices has also attracted cyberpunks and technophiles to exploit system vulnerabilities [49]. IoT devices are insecure, low-cost devices with a limited computational capacity. The availability of such devices in the novice user domain, unaware of security, has made the problem manifold. The IoT devices can easily be compromised to launch the most fatal distributed denial of service attacks (DDoS). DDoS attacks exhaust the system resources and make the system unavailable [48]. Though the work towards defense against IoT-generated DDoS attacks is in its inception, another perspective issue is deploying the computationally expensive intelligent model. It requires high-end resources and costly hardware. We have proposed a horizontally scalable architecture that is less



	0	1	2	3	4	5	6	7	Error	Rate
0	206	0	0	0	2	0	0	0	0.009615	2 / 208
1	0	512	1	0	0	0	1	0	0.003891	2 / 514
2	0	1	58	0	1	0	0	0	0.033333	2 / 60
3	0	0	0	268	1	0	0	0	0.003717	1 / 269
4	3	0	0	0	620	3	1	1	0.012739	8 / 628
5	0	0	0	0	2	85	0	0	0.022989	2 / 87
6	0	0	0	0	1	0	642	0	0.001555	1 / 643
7	0	0	0	0	2	0	0	41	0.046512	2 / 43
	209	513	59	268	629	88	644	42	0.008157	20 / 2,452

Confusion Matrix - Stack Ensemble : Test Set

(a)

	0	1	2	3	4	5	6	7	Error	Rate
0	1235	0	0	0	4	0	0	0	0.003228	4 / 1,239
1	0	3388	4	0	0	1	1	0	0.001768	6 / 3,394
2	0	2	399	0	2	0	0	0	0.009926	4 / 403
3	0	0	0	1733	1	0	0	0	0.000577	1 / 1,734
4	6	0	0	0	4162	3	1	1	0.002636	11 / 4,173
5	0	0	0	1	6	589	0	0	0.011745	7 / 596
6	0	0	0	0	1	0	4230	0	0.000236	1 / 4,231
7	0	0	0	0	5	0	0	264	0.018587	5 / 269
	1241	3390	403	1734	4181	593	4232	265	0.002432	39 / 16,039

Confusion Matrix - Stack Ensemble – Validation Set

(b)

	0	1	2	3	4	5	6	7	Error	Rate
0	1233	0	0	0	6	0	0	0	0.004843	6 / 1,239
1	0	3388	2	0	2	2	0	0	0.001768	6 / 3,394
2	0	1	399	0	3	0	0	0	0.009926	4 / 403
3	0	1	0	1732	1	0	0	0	0.001153	2 / 1,734
4	7	0	1	0	4158	6	1	0	0.003595	15 / 4,173
5	0	1	0	1	5	588	1	0	0.013423	8 / 596
6	0	0	0	0	9	1	4221	0	0.002364	10 / 4,231
7	0	0	1	0	7	0	0	261	0.02974	8 / 269
	1240	3391	403	1733	4191	597	4223	261	0.003679	59 / 16,039

Confusion Matrix - XGBoost – Validation Set

(c)

	0	1	2	3	4	5	6	7	Error	Rate
0	1233	0	0	0	6	0	0	0	0.004843	6 / 1,239
1	0	3388	2	0	2	2	0	0	0.001768	6 / 3,394
2	0	2	399	0	2	0	0	0	0.009926	4 / 403
3	0	0	0	1733	1	0	0	0	0.000577	1 / 1,734
4	16	22	0	0	4131	2	1	1	0.010065	42 / 4,173
5	0	17	0	0	5	574	0	0	0.036913	22 / 596
6	0	0	1	0	5	0	4225	0	0.001418	6 / 4,231
7	0	0	0	0	5	0	0	264	0.018587	5 / 269
	1249	3429	402	1733	4157	578	4226	265	0.005736	92 / 16,039

Confusion Matrix - DRF – Validation Set

(d)

	0	1	2	3	4	5	6	7	Error	Rate
0	1232	0	0	0	7	0	0	0	0.00565	7 / 1,239
1	0	3389	2	0	0	2	1	0	0.001473	5 / 3,394
2	0	0	401	0	2	0	0	0	0.004963	2 / 403
3	0	0	0	1733	1	0	0	0	0.000577	1 / 1,734
4	6	5	0	0	4157	4	1	0	0.003834	16 / 4,173
5	0	2	0	1	6	587	0	0	0.015101	9 / 596
6	0	0	0	0	8	0	4223	0	0.001891	8 / 4,231
7	0	0	0	0	3	0	0	266	0.011152	3 / 269
	1238	3396	403	1734	4184	593	4225	266	0.00318	51 / 16,039

Confusion Matrix - GBM – Validation Set

(e)

	0	1	2	3	4	5	6	7	Error	Rate
0	890	0	0	180	169	0	0	0	0.281679	349 / 1,239
1	0	3377	0	0	16	0	1	0	0.005009	17 / 3,394
2	0	1	381	0	21	0	0	0	0.054591	22 / 403
3	1	0	0	1435	298	0	0	0	0.172434	299 / 1,734
4	331	196	8	12	3616	2	0	8	0.133477	557 / 4,173
5	0	312	7	0	48	229	0	0	0.615772	367 / 596
6	0	6	0	0	64	0	4161	0	0.016545	70 / 4,231
7	218	0	0	0	23	0	0	28	0.895911	241 / 269
	1440	3892	396	1627	4255	231	4162	36	0.119833	1,922 / 16,039

Confusion Matrix - GLM – Validation Set

(f)

Fig. 16 Confusion Matrix for Base and Meta Models

Table 6 Base Models Hit Ratio

Device	XGBoost	DRF	GBM
Amazon Echo	0.996321	0.994264	0.99682
Belkin Wemo Motion Sensor	0.998753	0.999002	0.999065
Belkin Wemo Switch	0.999252	0.999564	0.999377
Insteon Camera	0.999377	0.999751	0.999564
NonIoT	0.999688	0.999813	0.999688
PIXSTAR Photo Frame	0.999751	0.999875	0.999938
Smart Things	0.999751	0.999938	1
Withings Aura Sleep Sensor	1	1	1

expensive than vertical scaling of the hardware. Furthermore, the framework is extensible, resilient, portable, and enabled for handling high-volume traffic. We recommend unfolding the proposed framework and applying it to the security problems.

6 Conclusion

The prolific increase in heterogeneous IoT devices and massive network traffic volume has tailpiece the operational onus to smart-environment operators and security enforcers. This paper has characterized the ingress of IoT traffic based on IoT devices' statistical and functional attributes. The pro-

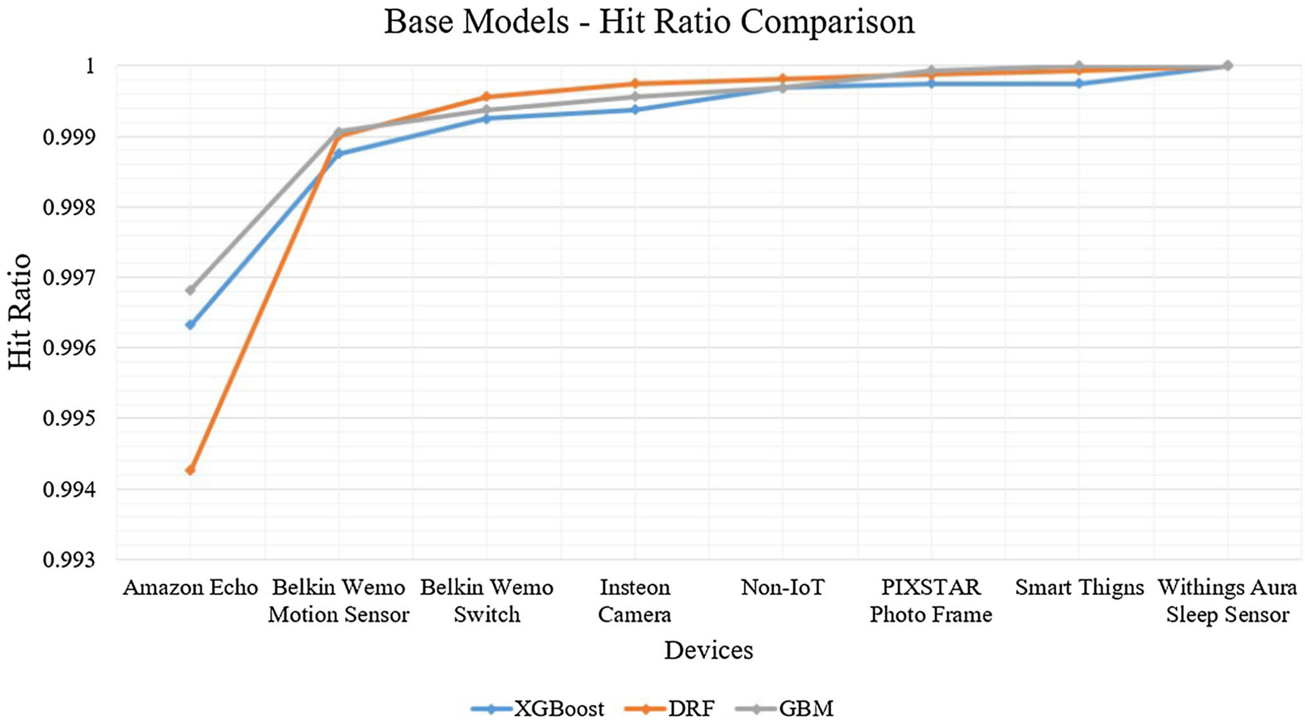


Fig. 17 Model Performance: Hit Ratio

Table 7 Stack Ensemble Performance Metrics

Device	Accuracy	Precision	Recall	F1-score
Amazon Echo	0.99937652	0.99677159	0.99516519	0.99596774
Belkin Wemo Motion Sensor	0.99950122	0.99823217	0.99941003	0.99882075
Belkin Wemo Switch	0.99950122	0.99007444	0.99007444	0.99007444
Insteon Camera	0.9998753	0.9994233	0.9994233	0.9994233
NonIoT	0.99812956	0.99736401	0.99545563	0.99640891
PIXSTAR Photo Frame	0.99931417	0.98825503	0.99325464	0.99074853
Smart Things	0.99981296	0.99976365	0.99952741	0.99964552
Withings Aura Sleep Sensor	0.99962591	0.98141264	0.99622642	0.98876404

Table 8 Base and Meta Models Performance Metric Comparison

Model	Accuracy	Precision	Recall	F1-score
Stack Ensemble	0.99939211	0.9939121	0.99606713	0.99498165
XGBoost	0.99908037	0.99164874	0.99494307	0.99326702
DRF	0.99855041	0.98928635	0.99375101	0.99147384
GBM	0.99920506	0.99441994	0.99631352	0.99536173
GLM	0.97004177	0.72807298	0.86856859	0.74825788

posed Stack Ensemble model stacks XGBoost, Distributed Random Forest, Gradient Boosting Machine, and General Linear Machine algorithms. Through experimental study, we demonstrated that the model outperformed with an accuracy of 99.94%. The paper has also illustrated the essential performance metrics (Precision, Recall, F1 score), model learning details, and comparative analysis of the stacking models from the experimentation study that many researchers usually missed. The higher values of Precision (99.4%), Recall (99.6%), and F1-score (99.5%) complement the system's accuracy. The article has also proposed a Docker container base scalable and distributed architecture for horizontal resource scaling for distributing the computational load. The novel framework is scalable, extensible to security solutions, and portable. The proposed framework is capable of handling high-volume real-time network traffic. The

Table 9 Comparison of state-of-the-art in IoT classification domain

Ref.	Objective	Method(s)	Dataset	Testbed configuration	Performance
[39]	IoT/Non-IoT Classification	Statistical analysis and packet inspection	Packet captures from [45]	NA	Accuracy: 98.7%
[45]	IoT/Non-IoT Classification in real-time	Multinomial Classification (Naive Bayes and Random Forest)	Living lab setup in UNSW University (28 IoT and Non-IoT captured for 6 months)	Smart Lab environment	Accuracy: 99.88%
[6]	Autonomous IoT Classify	Deep Learning	PCAP from 15 devices categorized into 4 classes	NA	Accuracy: 99.7%
[10]	Auto Classify IoT devices	Supervised Ensemble Learning	Packet captures from [45]	NA	Accuracy: 98.0% Recall: 99.2% Precision: 99.2%
[21]	Multi-class IoT Classify	Ensemble Learning	IoT traffic captures from Smart environments (Smart Grid, Smart Air Quality, Cyber Medical System)	NA	Accuracy: 91.7% (Binary), 90.6% (Multi-class)
[19]	Identification of IoT class (than IoT device)	Decision Tree, Adaboost, Logistic Regression, Random Forest	Packet captures from [45]	Split into 4 IoT Genres	Not specified
[13]	IoT Classification	Feature Ranking	15 IoT devices	In-Lab setup	Accuracy: 95.0%
[39]	IoT/Non-IoT classify	Payload Inspection	Packet captures from [45]	NA	Accuracy: 99.0%
[22]	IoT device classify	Deep Packet Inspection	Packet captures from [45]	NA	Not specified
[4]	IoT Classify	Payload and Flow features	IoT traces from 5 smart devices	Smart Home setup	Accuracy: 97.0%
[29]	IoT Classify	Featureset based on periodic communication	Packet captures from [45]	NA	Accuracy: 99.2% (33 devices)
Proposed Work	IoT/Non-IoT Classification in real-time	Stack Ensemble	Packet captures from [45]	In-Lab setup with Big-data tools	Accuracy: 99.94% Precision: 99.4% Recall: 99.6% F1-score: 99.5%

developed framework is evaluated on the Sydney dataset. However, it can further be evaluated on a variety of publically available benchmark IoT datasets.

In the future, it is recommended to integrate big-data tool(s) like Apache Spark due to its advanced analytics and in-memory processing capabilities. Furthermore, it is planned to unfold the framework architecture to apply cyber-attack issues and develop autonomous defense solutions.

Acknowledgements The authors would like to express their appreciation to anonymous reviewers for their insightful comments and helpful ideas.

Declarations

Conflict of interest The authors declare that they do not have any known personal relationship or competing financial interests that could have influenced the work reported in this paper.

References

1. Ahrenholz, J.: Comparison of CORE network emulation platforms. In: 2010-MILCOM 2010 Military Communications Conference, pp. 166–171. IEEE (2010). https://doi.org/10.1109/MILCOM.2010.5680218
2. Aksoy, A., Gunes, M.H.: Automated IoT Device Identification using Network Traffic. IEEE International Conference on Communications **2019-May** (2019). https://doi.org/10.1109/ICC.2019.8761559
3. Alippi, C., Ozawa, S.: Computational Intelligence in the Time of Cyber-Physical Systems and the Internet of Things. In: Artificial Intelligence in the Age of Neural Networks and Brain Computing, pp. 245–263. Elsevier (2019). https://doi.org/10.1016/B978-0-12-815480-9.00012-8
4. Ammar, N., Noirie, L., Tixeuil, S.: Autonomous Identification of IoT Device Types based on a Supervised Classification. IEEE International Conference on Communications **2020-June** (2020). https://doi.org/10.1109/ICC40277.2020.9148821
5. Apache Software Foundation: Apache Kafka. (Accessed on 2021-05-28) (2017). https://kafka.apache.org/
6. Bai, L., Yao, L., Kanhere, S.S., Wang, X., Yang, Z.: Automatic device classification from network traffic streams of internet of things. In: 2018 IEEE 43rd conference on local computer networks (LCN), pp. 1–9. IEEE (2018)
7. Baker, F., Foster, B., Sharp, C.: Cisco Architecture for Lawful Intercept in IP Networks-RFC3924. (Accessed on 2021-05-20) (2004). https://datatracker.ietf.org/doc/html/rfc3924
8. Bao, J., Hamdaoui, B., Wong, W.K.: IoT Device Type Identification Using Hybrid Deep Learning Approach for Increased IoT Security. 2020 International Wireless Communications and Mobile Computing, IWCMC 2020 pp. 565–570 (2020). https://doi.org/10.1109/IWCMC48107.2020.9148110
9. Chamola, V.; Hassija, V.; Gupta, V.; Guizani, M.: A comprehensive review of the COVID-19 pandemic and the role of IoT, Drones, AI, Blockchain, and 5G in Managing its Impact. IEEE Access **8**(April), 90225–90265 (2020). https://doi.org/10.1109/ACCESS.2020.2992341
10. Cirillo, G.; Passerone, R.: Packet length spectral analysis for IoT flow classification using ensemble learning. IEEE Access **8**, 138616–138641 (2020). https://doi.org/10.1109/ACCESS.2020.3012203
11. Čolaković, A.; Hadžialić, M.: Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. Comput. Netw. **144**, 17–39 (2018). https://doi.org/10.1016/j.comnet.2018.07.017
12. Dainotti, A.; Pescapé, A.; Claffy, K.C.: Issues and future directions in traffic classification. IEEE Netw. **26**(1), 35–40 (2012). https://doi.org/10.1109/MNET.2012.6135854
13. Desai, B.A., Divakaran, D.M., Nevat, I., Peter, G.W., Gurusamy, M.: A feature-ranking framework for IoT device classification. 2019 11th International Conference on Communication Systems and Networks, COMSNETS 2019 **2061**, 64–71 (2019). https://doi.org/10.1109/COMSNETS.2019.8711210
14. DEVNET, C.: Cisco DevNet Code Exchange: Discover code repositories related to Cisco technologies. (Accessed on 2021-05-30) (2016). https://developer.cisco.com/codeexchange/github/repo/cisco/joy/
15. Dulhare, U.N., Mubeen, A., Ahmad, K.: Hands On H2O Machine Learning Tool. In: Machine Learning and Big Data, pp. 423–453. Wiley (2020). https://doi.org/10.1002/9781119654834.ch15
16. Guo, H.; Heidemann, J.: Detecting IoT devices in the internet. IEEE/ACM Trans. Netw. **28**(5), 2323–2336 (2020). https://doi.org/10.1109/TNET.2020.3009425
17. Hamza, A., Ranathunga, D., Gharakheili, H.H., Roughan, M., Sivaraman, V.: Clear as MUD: Generating, validating and applying IoT behavioral profiles. IoT S and P 2018-Proceedings of the 2018 Workshop on IoT Security and Privacy, Part of SIGCOMM 2018 pp. 8–14 (2018). https://doi.org/10.1145/3229565.3229566
18. Holst, A.: Internet of Things (IoT) connected devices worldwide from 2019 to 2030. (Accessed on 2021-05-12) (2021). https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/
19. Hsu, A., Tront, J., Raymond, D., Wang, G., Butt, A.: Automatic IoT Device Classification using Traffic Behavioral Characteristics. Conference Proceedings-IEEE SOUTHEASTCON **2019-April**, 1–7 (2019). https://doi.org/10.1109/SoutheastCon42311.2019.9020640
20. Izonin, I., Tkachenko, R., Vitynskyi, P., Zub, K., Tkachenko, P., Dronyuk, I.: Stacking-based GRNN-SGTM Ensemble Model for Prediction Tasks. In: 2020 International Conference on Decision Aid Sciences and Application (DASA), pp. 326–330. IEEE (2020). https://doi.org/10.1109/DASA51403.2020.9317124
21. Khadse, V.; Mahalle, P.N.; Shinde, G.R.: A novel approach of ensemble learning with feature reduction for classification of binary and multiclass IoT data. Turkish J. Comput. Math. Educ. (TURCOMAT) **12**(6), 2072–2083 (2021). https://doi.org/10.17762/turcomat.v12i6.4811
22. Khandait, P.; Hubballi, N.; Mazumdar, B.: IoTHunter: IoT network traffic classification using device specific keywords. IET Netw. **10**(2), 59–75 (2021). https://doi.org/10.1049/ntw2.12007
23. Kraus, M.; Feuerriegel, S.; Oztekin, A.: Deep learning in business analytics and operations research: Models, applications and managerial implications. Eur. J. Op. Res. **281**(3), 628–641 (2020). https://doi.org/10.1016/j.ejor.2019.09.018
24. Laner, M.; Nikaein, N.; Svoboda, P.; Popovic, M.; Drajic, D.; Krco, S.: Traffic models for machine-to-machine (M2M) communications. Elsevier Ltd, New York (2015). https://doi.org/10.1016/b978-1-78242-102-3.00008-3
25. Lear, E., Droms, R., Romascanu, D.: Manufacturer Usage Description Specification. (Accessed on 2021-05-23) (2018). https://tools.ietf.org/id/draft-ietf-opsawg-mud-22.html
26. Lee, J.; Bagheri, B.; Kao, H.A.: A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems. Manuf. Lett. **3**, 18–23 (2015). https://doi.org/10.1016/j.mfglet.2014.12.001

27. Lemley, J.; Bazrafkan, S.; Corcoran, P.: Deep learning for consumer devices and services: Pushing the limits for machine learning, artificial intelligence, and computer vision. IEEE Consumer Electron. Mag. **6**(2), 48–56 (2017). https://doi.org/10.1109/MCE.2016.2640698
28. Lim, Y.s., Kim, H.c., Jeong, J., Kim, C.k., Kwon, T.T., Choi, Y.: Internet traffic classification demystified. In: Proceedings of the 6th International COnference on-Co-NEXT-10, p. 1. ACM Press (2010). https://doi.org/10.1145/1921168.1921180
29. Marchal, S.; Miettinen, M.; Nguyen, T.D.; Sadeghi, A.R.; Asokan, N.: AuDI: Toward autonomous IoT device-type identification using periodic communication. IEEE J. Selected Areas Commun. **37**(6), 1402–1412 (2019). https://doi.org/10.1109/JSAC.2019.2904364
30. Mavrogiorgou, A., Kiourtis, A., Kyriazis, D.: IoT Devices Recognition through Object Detection and Classification Techniques. In: 2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4), pp. 12–20. IEEE (2019). https://doi.org/10.1109/WorldS4.2019.8903926
31. Meidan, Y., Bohadana, M., Shabtai, A., Guarnizo, J.D., Ochoa, M., Tippenhauer, N.O., Elovici, Y.: ProfilIoT: A Machine Learning Approach for IoT Device Identification Based on Network Traffic Analysis. In: Proceedings of the Symposium on Applied Computing, vol. Part F1280, pp. 506–509. ACM (2017). https://doi.org/10.1145/3019612.3019878
32. Mendes Junior, J.J.A., Freitas, M.L., Siqueira, H.V., Lazzaretti, A.E., Pichorim, S.F., Stevan, S.L.: Feature selection and dimensionality reduction: An extensive comparison in hand gesture classification by sEMG in eight channels armband approach. Biomed. Signal Process. Control **59**, 101920 (2020). https://doi.org/10.1016/j.bspc.2020.101920
33. Mills, D., Delaware, U., J. Martin, E., Burbank, J., Kasch, W.: RFC 5905-Network Time Protocol Version 4: Protocol and Algorithms Specification. (Accessed on 2021-06-10) (2010). https://datatracker.ietf.org/doc/html/rfc5905
34. Moore, A.W.; Zuev, D.: Internet traffic classification using bayesian analysis techniques. Perform. Eval. Rev. **33**(1), 50–60 (2005). https://doi.org/10.1145/1071690.1064220
35. Ndiaye, M.; Oyewobi, S.S.; Abu-Mahfouz, A.M.; Hancke, G.P.; Kurien, A.M.; Djouani, K.: IoT in the wake of COVID-19: a survey on contributions. Challenges and Evolution. IEEE Access **8**, 186821–186839 (2020). https://doi.org/10.1109/ACCESS.2020.3030090
36. Nguyen, T.T.; Armitage, G.: A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surveys Tutor. **10**(4), 56–76 (2008). https://doi.org/10.1109/SURV.2008.080406
37. Pacheco, F.; Exposito, E.; Gineste, M.; Baudoin, C.; Aguilar, J.: Towards the deployment of machine learning solutions in network traffic classification: a systematic survey. IEEE Commun. Surveys Tutor. **21**(2), 1988–2014 (2019). https://doi.org/10.1109/COMST.2018.2883147
38. Reynolds, J., Postel, J.: ASSIGNED NUMBERS. (Accessed on 2021-05-23) (1992). https://datatracker.ietf.org/doc/html/rfc1340
39. Santos, M.R., Andrade, R.M., Gomes, D.G., Callado, A.C.: An efficient approach for device identification and traffic classification in IoT ecosystems. Proceedings-IEEE Symposium on Computers and Communications **2018-June**, 304–309 (2018). https://doi.org/10.1109/ISCC.2018.8538630
40. Shafiq, M., Yu, X., Laghari, A.A., Yao, L., Karn, N.K., Abdessamia, F.: Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms. In: 2016 2nd IEEE International Conference on Computer and Communications (ICCC), vol. 10, pp. 2451–2455. IEEE (2016). https://doi.org/10.1109/CompComm.2016.7925139
41. Sisinni, E.; Saifullah, A.; Han, S.; Jennehag, U.; Gidlund, M.: Industrial Internet of Things: challenges, opportunities and directions. IEEE Trans. Ind. Inf. **14**(11), 4724–4734 (2018). https://doi.org/10.1109/TII.2018.2852491
42. Sivanathan, A., Gharakheili, H.H.: IoT Security-IoT Traffic Analysis. (Accessed on 2021-05-26) (2018). https://iotanalytics.unsw.edu.au/iottraces
43. Sivanathan, A.; Gharakheili, H.H.; Loi, F.; Radford, A.; Wijenayake, C.; Vishwanath, A.; Sivaraman, V.: Classifying IoT devices in smart environments using network traffic characteristics. IEEE Trans. Mobile Comput. **18**(8), 1745–1759 (2019). https://doi.org/10.1109/TMC.2018.2866249
44. Sivanathan, A., Gharakheili, H.H., Sivaraman, V.: Can We Classify an IoT Device using TCP Port Scan? 2018 IEEE 9th International Conference on Information and Automation for Sustainability, ICIAfS 2018 (2018). https://doi.org/10.1109/ICIAFS.2018.8913346
45. Sivanathan, A., Sherratt, D., Gharakheili, H.H., Radford, A., Wijenayake, C., Vishwanath, A., Sivaraman, V.: Characterizing and classifying IoT traffic in smart cities and campuses. In: 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), vol. 18, pp. 559–564. IEEE (2017). https://doi.org/10.1109/INFCOMW.2017.8116438
46. Snehi, J.; Bhandari, A.; Baggan, V.; Snehi, M.: Diverse methods for signature based intrusion detection schemes adopted. Int. J. Recent Technol. Eng. (IJRTE) **9**(2), 44–49 (2020)
47. Snehi, J., Bhandari, A., Snehi, M., Tandon, U., Baggan, V.: Global intrusion detection environments and platform for anomaly-based intrusion detection systems. In: Proceedings of Second International Conference on Computing, Communications, and Cyber-Security, pp. 817–831. Springer (2021)
48. Snehi, M., Bhandari, A.: Apprehending Mirai Botnet Philosophy and Smart Learning Models for IoT-DDoS Detection. In: 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 501–505 (2021). https://doi.org/10.1109/INDIACom51348.2021.00089
49. Snehi, M.; Bhandari, A.: Vulnerability retrospection of security solutions for software-defined cyber-physical system against DDOS and IoT-DDOS attacks. Comput. Sci. Rev. (2021). https://doi.org/10.1016/j.cosrev.2021.100371
50. Tkachenko, R.; Izonin, I.; Dronyuk, I.; Logoyda, M.; Tkachenko, P.: Recovery of missing sensor data with GRNN-based cascade scheme. Int. J. Sensors, Wireless Commun. Control **11**(5), 531–541 (2021). https://doi.org/10.2174/2210327910999200813151904
51. Trevisan, M., Drago, I., Mellia, M., Munafo, M.M.: Towards web service classification using addresses and DNS. In: 2016 International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 38–43. IEEE (2016). https://doi.org/10.1109/IWCMC.2016.7577030
52. Verma, J., Bhandari, A., Singh, G.: A meta-analysis of role of network intrusion detection systems in confronting network attacks. In: 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), pp. 506–511 (2021). https://doi.org/10.1109/INDIACom51348.2021.00090
53. Weiss, M.A.; Eidson, J.; Barry, C.; Broman, D.; Iannucci, B.; Lee, E.A.; Stanton, K.; Sr., Goldin, L.: Time-aware applications, computers and communication systems (TAACCS). Tech. rep. (2015). https://doi.org/10.6028/NIST.TN.1867
54. Yu, T., Sekar, V., Seshan, S., Agarwal, Y., Xu, C.: Handling a trillion (unfixable) flaws on a billion devices. In: Proceedings of the 14th ACM Workshop on Hot Topics in Networks, pp. 1–7. ACM (2015). https://doi.org/10.1145/2834050.2834095