Energy-Efficient Tactile-driven Rule Configuration and Anomaly Detection in Industrial IoT Systems

Lizhuang Tan, Amritpal Singh, Wei Zhang, Hongjuan Pei, Peiying Zhang,
Prabhjot Kaur Chahal, Maninderpal Singh

***Abstract*—The Industrial Internet of Things (IIoT) enables communication among automation systems, machinery, and sensors in an industrial setting. To optimize critical industrial operations, a substantial volume of data concerning diverse in-factory activities and automation services is generated by IoT devices and sensors. This data is subsequently transferred to distant processing systems for analysis and decision-making. Nevertheless, a substantial latency in data transmission or any abnormality in the generated data may result in delayed or erroneous decisions, consequently impacting the efficacy of essential industrial systems. To address these challenges, we established an intelligent network architecture utilizing software-defined networking that achieves tactile latencies efficiently while handling industrial data traffic in an energy-efficient manner. To address the initial challenge, the suggested architecture utilizes the Self-Organized Maps approach to distinguish between industrial traffic requiring tactile latencies and non-tactile traffic. We utilize a binary tree-based flow table mapping method to enhance flow table matching and decrease lookup times. To address the second challenge, we employ the Support Vector Machine technique to identify anomalies in real-time industrial data traffic. The Hadoop system and Mininet emulator are utilized to evaluate the proposed architecture using the UNSW dataset. The results demonstrate the effectiveness of the suggested solution in providing energy-efficient tactile assurances and identifying anomalies in traffic.**

***Index Terms*—Industrial IoT, Internet of Things, Smart City, Tactile Network, Traffic Anomalies.**

I. Introduction

The Internet of Things (IoT) has significantly grown and transformed the global infrastructure landscape. Recently, the IoT has experienced a substantial transformation and challenge due to the emergence of various disruptive concepts and technologies. Within this framework, Industrial IoT (IIoT) has arisen as a pivotal facilitator in the establishment of automated industries and smart factories, driven by recent advancements such as Artificial Intelligence (AI). The IIoT comprises a network of interconnected devices and sensors utilized within an industrial setting. The major objective of these devices is to incessantly collect, disseminate, and assess critical data [1]. Integrated sensors in industrial machinery collect diverse data, including important and sensitive information, which might impact decision-making for various intelligent applications [2]. Figure 1 illustrates a standard IIoT setup in which sensors gather data and relay it to a distant processing system. A cloud-based big data processing cluster is used to analyse data for essential objectives in industrial systems and product quality monitoring applications. This data must be transmitted swiftly and reliably to processing systems [3]. Nonetheless, it necessitates the swift and reliable dissemination of collected data and the prompt and precise processing of the substantial data produced by industrial sensors [4].

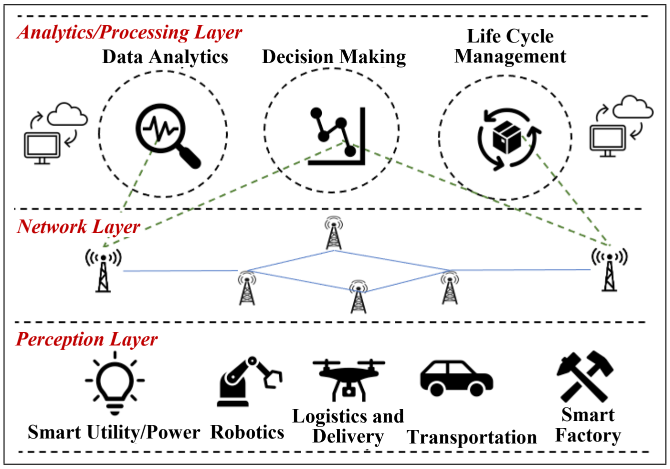


Fig. 1: A Typical Scenario for IIoT Applications

This scenario encounters two primary problems: a) transmission delay and b) abnormalities in data traffic. These two issues undermine any proposed solution in industrial networks. The challenge with the former is that the generated data must be delivered seamlessly and securely to the essential decision-making system. Similarly, the latter issue deals with data abnormalities, which might result in poor quality decisions.

To address the issue of data transfer latency, it is essential

L. Tan and W. Zhang are with the Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Ji'nan 250014, China. L. Tan is also with Department of Computer Science and Engineering, Pohang University of Science and Technology, Pohang, 37673, Korea. Email: tanlzh@sdas.org and wzhang@sdas.org

A. Singh is with Department of Computer and Information Sciences, Northumbria University, UK. Email: amritpal2.singh@northumbria.ac.uk

H. Pei is with the School of Engineering Science, University of Chinese Academy of Sciences, Beijing, 100049, China. Email: peihongjuan@ucas.ac.cn

P. Zhang is with the Qingdao Institute of Software, College of Computer Science and Technology, China University of Petroleum (East China), Qingdao 266580, China and also with the Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China Email: zhangpeiying@upc.edu.cn

P. K. Chahal is with the CEC, Chandigarh Group of Colleges, Mohali, Punjab, India. Email: prabh0480@gmail.com

M. Singh is with the Electrical and Computer Engineering Department, Southern Methodist University, Dallas, USA. Email: maninderpals@smu.edu

Corresponding Authors: Hongjuan Pei and Peiying Zhang

to identify an appropriate underlying network technology that guarantees ultra-low latency, while also ensuring high availability, and reliability for the transmission of data acquired by industrial sensors [5]. The International Telecommunication Union (ITU) designated the name Tactile Internet (TI) to describe an innovative Internet network capable of delivering ultra-low latency while ensuring high availability and reliability [6]. Nonetheless, not all created data require tactile assurances and can be managed on a best-effort basis. It is essential to comprehend the data features and ascertain their Quality of Service (QoS) needs for dynamic management [7]. This can enhance the utilization of network resources that would otherwise be maximally employed to ensure tactile assurances for incoming data packets. Software-defined Networking (SDN) appears to be ideally suited for managing industrial data gathered via sensors, providing tactile assurances while enabling network operators to optimize resource utilization through dynamically configurable methods implemented at the control layer [8]–[10]. This work proposes the creation of a TI-driven Software-defined industrial data-sharing system that employs an AI-based approach to dynamically classify incoming data traffic according to its QoS needs (tactile or non-tactile). Subsequently, the flow tables on the forwarding devices are optimized by reorganizing them with a binary tree data structure to align appropriate flow entries with the incoming differentiated data traffic.

Assuming that the industrial data traffic exhibit anomalous behavior (due to any attack or faulty sensor), it will adversely affect data quality; the acquired data may be erroneous, inaccurate, or incomplete, rendering it unsuitable for confident decision-making [11]. Thus, monitoring and assessing the essential parameters of the generated data becomes crucial. Thus, we need to have a near real-time system that enables the rapid identification of abnormalities or variations from standard operations that may signify safety concerns [12], [13]. If traffic irregularities are recognized swiftly, it becomes feasible to adjust the proposed solution to enhance accuracy and performance [14].

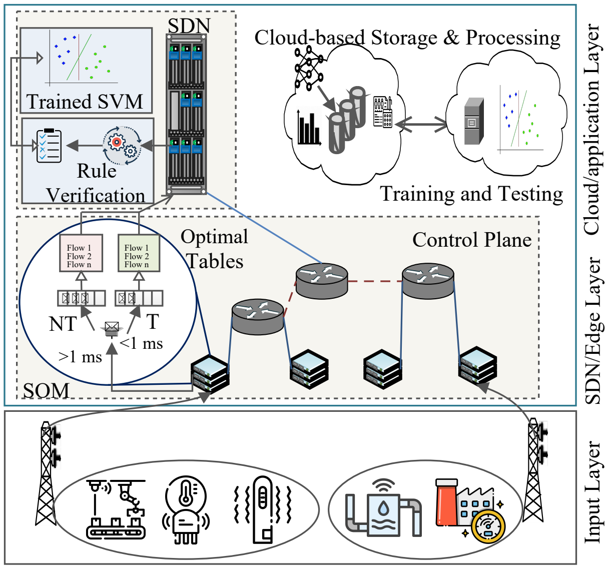
Numerous researchers have suggested methods to address the aforementioned challenges [15], [16]. Nevertheless, the majority of these solutions are energy-intensive and do not satisfy the criteria of the UN Sustainable Development Goals 2030 regarding Net Zero[1]. Energy considerations in network data transmission and anomaly detection are essential for maximizing operational efficiency and cost-effectiveness. The development of the energy-aware data transmission methods significantly increased energy utilisation. Moreover, legislative frameworks pertinent to energy management, such as the European Union's Ecodesign Directive[2], impose stringent energy efficiency standards, compelling network operators to use more sustainable methods. Effective energy management strategies and standards facilitate scalability, enabling the growth of network infrastructure without a corresponding rise in energy usage. Consequently, incorporating energy efficiency into network data processing is essential. This is a crucial method for cost reduction, environmental effect mitigation, and adherence to changing requirements.

So, the following contributions are provided in this paper:

- We design an energy-efficient data-sharing mechanism to ensure tactile guarantees while simultaneously optimizing network resource usage.
- We design an SVM-based approach to identify anomalous traffic in the underlying networks while transmitting enormous amounts of IIoT data.

II. System Model

The proposed system model, as shown in Fig. 2, is subdivided into three layers that are discussed below.



NT: Non Tactile, T: Tactile

Fig. 2: System Model

A. Input Layer

This layer comprises different IoT devices and sensors deployed within industry setup for provisioning various monitoring and control services.

B. SDN/Edge Layer

The network topologies are dynamic; hence, SDN deals with these challenges while sending data, adapting to changing topology effectively and optimizing QoS to ensure timely and reliable delivery of data packets. The SDN layer also acts as an edge layer to execute computational operations. The control plane of SDN includes the network components responsible for carrying data across the network; this includes SDN switches and routers. When an SDN switch receives a packet for forwarding, it checks whether an existing flow rule is available. A request is generated and sent to the SDN controller for a new flow rule in the non-availability of a flow rule. However, flow rules differ for different QoS requirements of incoming packets; the proposed model uses SOM-based

[1] https://sdgs.un.org/2030agenda

[2] https://commission.europa.eu/energy-climate-change-environment

classifiers deployed on SDN switches. These classifiers segregate incoming packets into two different queues based on the QoS requirements of the packet. Deployed rule checks for the QoS delay requirements to be either less than 1 ms or greater; packets with a QoS requirement of less than 1 ms are sent to a queue of tactile packets and others to a non-tactile packets queue. Accordingly, the different flow rules for tactile and non-tactile packets are fetched from the SDN controller into the SDN switch's flow table for forwarding data packets. If, in a case, no flow rule is matched, then a request is sent to the controller. On receiving a request for a new flow rule, the SDN controller creates new flow rules based on the pre-installed protocols.

C. Cloud/Application Layer

The cloud/application layer is the central data processing and analysis hub where all AI models are trained. IIoT data is processed using SVM over a MapReduce-based big data processing cluster for efficient feature extraction and decision-making.

III. Data Dissemination Approach for Tactile Applications

The energy consumption is one of the most important factors affecting operational efficiency and sustainability in tactile applications of IIoT. Classifying IIoT applications based on tactile and non-tactile interactions offers a tactical structure for maximizing resource distribution, especially concerning energy usage. The operational needs of tactile applications necessitate more significant energy inputs due to their requirement for real-time engagement and prompt response in operating machines and processes.

The proposed energy-efficient data dissemination mechanism uses the Self-organized Map (SOM) to differentiate the IIoT-based applications and classify them into tactile and non-tactile categories. In the proposed scheme, the binary tree data structure is also integrated to divide the flow tables installed on the forwarding devices to include flow entries relevant to the tactile and non-tactile traffic. Finally, a mapping scheme tries to organise the incoming traffic into an appropriate sub-flow table for matching and action. The proposed scheme is elaborated in the subsequent sub-section.

A. Application-specific Classification using SOM

The incoming traffic from IIoT-based application can be divided into groups based on the significance of each application in terms of tactile assurances. The payload type, port number, and other attributes allow for the differentiation of incoming packets. The proposed strategy relies on the incoming traffic classification in Table I.

TABLE I: Application-Specific Classification Based on Latency

Latency	Tentative Applications	Category
$\leq$ 1ms	Smart Manufacturing, Industrial Security, Energy Management, Industrial Automation, Remote Monitoring	Tactile-Based Services
$>$ 1ms	Other applications	Non-Tactile-based Services

To improve the QoS for IIoT-based applications, SOM, based on an Artificial Neural Network (ANN), divides incoming traffic into two distinct classes: tactile and non-tactile. Consequently, the approaching traffic forms two lineups. i.e., $\mathbb{Q}_{\mathbb{TS}}$ (Tactile Service) and $\mathbb{Q}_{\mathbb{NTS}}$ (Non-tactile Service).

- **Training:** In this phase, we generate a vector that responds uniformly to a homogeneous traffic pattern. Initially, the neurons' weights are assigned by using an eigenvector to ignite the iteration process. SOM is an unsupervised learning model that clusters the unlabeled dataset for training the model. The SOM model maintains the topological properties of the collected input by using competitive learning methods.
- **Mapping:** Initially, the random weights are assigned to the neurons to open the training process. In the mapping phase, the weight vector searches for the best weight matches the sample. In the next phase, the neighboring weights that match the selected weight vector are rewarded with the same weight to create a cluster of similar weights (properties) using the Euclidean distance and finalizing the best machine unit.

B. Binary Tree-based Flow Table Optimization

Binary tree is a data structure that stores the data blocks hierarchically [17]. It is a non-linear data storage approach where the stored data is neither linear nor sequential. The working structure of the binary search tree is shown in Fig. 3.

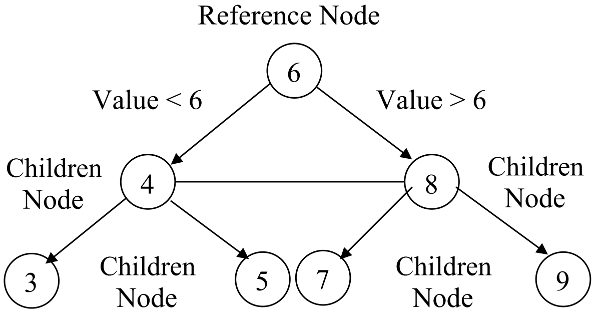
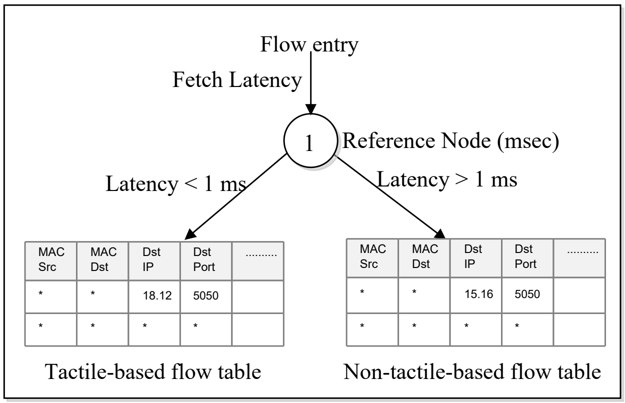


Fig. 3: Binary Tree Structure

In the binary tree structure, the leaf node (reference node) stores the key value of the children node and the children node stores the actual data. A tree is binary only if a leaf node contains two child nodes and similar patterns at various levels. In the next level, if the key value of the leaf node is less, then the data is stored on the left node, otherwise on the right node.

In the proposed scheme, the binary tree is used to manage the flow table entries to improve the lookup speed while matching the flow table entries with the generated flow rules. The binary tree storing the flow table entries provides manifold benefits, such as application-based indexing, direct and sequential searching, and faster data insertion and retrieval. Here, only two flow tables are stored on the configured devices: tactile and non-tactile. The tactile-based flow table matches the packets (Tactile-based services in Table I) with the flow table entries to provide latency-free services. On the other hand, the packet flow rules (Non-tactile services in Table I) match the Non-tactile flow table for data forwarding. In the proposed scheme, the critical value of the leaf node is assigned considering the latency metric, i.e., the time taken to transfer the data/packets from one end to another. According to **ITU-T Technology Watch Report** [18], the maximum latency a tactile network may tolerate is 1 ms. Therefore, 1ms latency is considered a threshold value while segregating the flow entries in the two defined flow tables (Tactile-based and non-tactile-based), as demonstrated in Fig. 4.



MAC Src	MAC Dst	Dst IP	Dst Port	....
*	*	18.12	5050	
*	*	*	*	

MAC Src	MAC Dst	Dst IP	Dst Port	....
*	*	15.16	5050	
*	*	*	*	

Fig. 4: Flow Table Optimization using Binary Tree

C. Application-based Tactile-driven Mapping Scheme

In this scheme, the flow table entries are segregated and stored into two flow tables depending upon the priority of the incoming packets from a particular application. The scheme helps to configure the low-latency tactile network. The Algorithm 1 shows the proposed application-specific data-sharing workflow and flow table optimization mechanism. The incoming traffic from various applications is mapped with the SOM model to fetch the priority of the application, and accordingly, the Tactile-based ($\mathbb{S}_{\mathbb{T}}$) or Non-tactile-based services ($\mathbb{S}_{\mathbb{NT}}$) are provided (lines 1-9). If the application is eligible for $\mathbb{S}_{\mathbb{T}}$, the flow rule of the application packets is mapped with the Tactile-based flow table ($\mathbb{FT}_{\mathbb{T}}$) and maintains the quality of the service. In another case, the packet flow rule matches the normal flow table (Non-tactile-based) ($\mathbb{FT}_{\mathbb{NT}}$) for flow forwarding. The time, packet reach to the destination, and packet latency is calculated for futuristic purposes (line 10-20). If the calculated latency is less than 1 msec and the application is eligible for $\mathbb{S}_{\mathbb{T}}$, existing flow entry ($\mathbb{FE}$) latency is compared. If the existing $\mathbb{FE}$ latency exceeds the current flow rule latency, the existing $\mathbb{FE}$ is replaced with the current formulated $\mathbb{FE}$ that leads towards tactile network expectations. If a $\mathbb{FE}$ for an application does not exist in the $\mathbb{FT}_{\mathbb{T}}$, a new entry is stored in the $\mathbb{FT}_{\mathbb{T}}$ (lines 21-31).

IV. Real-time Anomaly Detection

The dynamic growth of the network due to the configuration of network devices worldwide resulted in an increase in traffic flow rate over the underlying networks. With the surge in traffic, the rate of malicious attacks and anomalies also increased [19]. Therefore, a model is required for a preliminary inspection of the incoming traffic and discarding the anomalous traffic from the network. Therefore, SVM, an ML model, is integrated with the underlying network for anomalous traffic detection. The SVM is a supervised learning model to analyze the data for classification and regression analysis.

Algorithm 1 Application-specific Data Sharing Mechanism

```
Input: Drone Traffic Packet: $\mathbb{P}$
Output: Matched $\mathbb{FE}$
 1: for i $\neq$ NULL do
 2:     for j $\neq$ NULL do
 3:         Assign random weights to neurons: $\mathbb{W}_{ij} \in (0, 1)$
 4:         Target input vector: $\mathbb{D}_t$
 5:         Calculate $\mathbb{D}_{(i,j)} = \sqrt{\sum_i (y_i - x_i)^2}$
 6:         BMU $\leftarrow$ MIN($\mathbb{D}_{(i,j)}$)
 7:         Reiterate BMU neighbourhood weights
 8:         Mapping: $\mathbb{P}_i \in (\mathbb{S}_{\mathbb{T}}, \mathbb{S}_{\mathbb{NT}})$        ▷ $\mathbb{P} \rightarrow$ Packet
 9:     end for
10:     if $\mathbb{P}_i \in \mathbb{S}_{\mathbb{T}}$ then
11:         Fetch: $\mathbb{FT}_{\mathbb{T}}$
12:         Mapping: Flow rule $\rightleftarrows$ Flow table
13:         Packet forwarding to destination
14:         Calculate the latency of the service ($\mathbb{L}_i$)
15:     else
16:         Fetch: $\mathbb{FT}_{\mathbb{NT}}$
17:         Mapping: Flow rule $\rightleftarrows$ Flow table
18:         Packet forwarding to destination
19:         Calculate the latency of the service ($\mathbb{L}_i$)
20:     end if
21:     if $\mathbb{L}_i \leq 1$ & $i \in \mathbb{S}_{\mathbb{T}}$ then
22:         Fetch $\mathbb{FE} \in \mathbb{P}_i$ from $\mathbb{FT}_{\mathbb{T}}$
23:         if $\mathbb{FE} \neq NULL$ then
24:             Fetch $\mathbb{L}_{old}$ from $\mathbb{FE}$        ▷ $\mathbb{L}_{old} \rightarrow$ Old Latency
25:             if $\mathbb{L}_{old} > \mathbb{L}_i$ then
26:                 Override: $\mathbb{FE}_{old}$ with $\mathbb{FE}_i$
27:             end if
28:         else
29:             Insertion: $\mathbb{FT}_i$ into $\mathbb{FT}_{\mathbb{T}}$
30:         end if
31:     end if
32: end for
```

Initially, the model is trained using the appropriate dataset, and accordingly, the provided input is classified into defined classes. The considered dataset is divided into 80:20 ratios for training and testing purposes. The 80% data of the dataset is used for training purposes, and the rest 20% data is for testing purposes. The training and testing of the model are discussed in the below-mentioned points.

- The training dataset of n points is given as:

$$(x_1, y_1), \ldots, (x_n, y_n) \tag{1}$$

  where, y_i indicates the extracted output in-terms of labels (1, -1) for the considered x_i data points.
- The nonidentical data points must be separated and divided into desired classes for identification purposes. The positioning of the data points is decided as per the values extracted by the given equations:

$$w^T - b \geq 0 \tag{2}$$

$$w^T - b < 0 \tag{3}$$

where, w is defined as $vectors(w_0, w_1, w_2, \ldots, w_m)$, T is the summation of different dimensional data points, and b is the biased value. If the sum of the projection of the data point and vector and biased value is greater than or equal to 0, the data point is labelled as 1, as shown in Eq. (2), and if the sum of the projection of the data point and vector and biased value is less than 0, labelled as -1 shown in Eq. (3) and accordingly the data points are positioned in the plane at initial stages.

- To improve the model's accuracy, there is a need to maximize the margins (d) between the data points. The d is calculated by using following equation.

$$d = \frac{2}{||w||} \tag{4}$$

To maximize the distance/margin between two nonidentical data points, there is a need to minimize the value of w and in-resultant a distinct hyperplane is formulated to classify the objects into defined classes.

- Finally, the trained model is tested using the 20% dataset to check the accuracy of the proposed model.
- Further, the trained model is tuned to improve the accuracy of the proposed model.

The trained model is configured in the underlying network for anomalous traffic detection, and the anomalous traffic is halted for further processing. The detailed working of the proposed model is shown in Fig. 5.

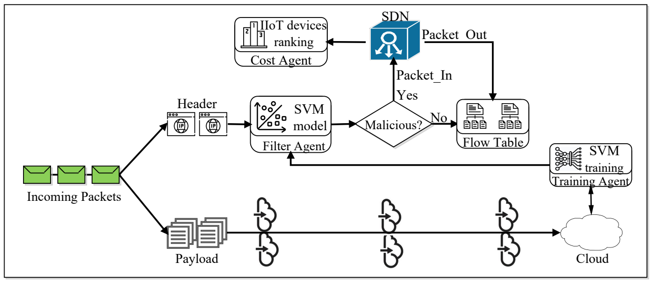


Fig. 5: Adaptive SVM-based Anomaly Detection

The step-wise working of the model is mentioned in the Algorithm 2. The incoming packet from the configured devices is initially forwarded to the packet analyzer. The packet analyzer fetches the packet header and payload from the sent packet (Step 1-4). The fetched header is forwarded to the trained SVM model to analyse the signature of the incoming traffic from various connected input devices. The trained SVM model labels the header as per the model's classification (Normal: 1, Anomalous: -1) (Step 5). If the header's label is -1, the *Packet_In* request is generated and redirected towards the configured SDN controller for further processing. The SDN controller creates a new flow rule to cease the respective traffic from the network. The SDN controller generates *Packet_Out* request to implement the generated flow rule on the configured switches in the network. In continuation, the SDN controller forwards a packet to the *Cost Agent* to update the ranking of the input device (Ranking) for futuristic purposes (Step 6-10). Initially, the ranking of the input devices is considered as 1 and with the flow of the traffic and type of traffic generated by the input devices, the ranking of the same is updated.

In case, the label is 1, the request is forwarded to the respective flow table to fetch the flow rule entry that helps to redirect the traffic to the designated switch (Step 12-13). The packet analyzer initially segregated the packet into header and payload. The header is used to identify the type of traffic forwarded by the input device. At the same time, the fetched payload of the traffic is forwarded to the cloud framework. The signature updation of the anomalous traffic is a dynamic process. Therefore, an adaptive framework is required to update the system by updating the signatures of the anomalous traffic. To consider the same, the Age of Information (AoI) concept is used for periodic training of the SVM model. The threshold value to train the model depends upon the traffic flow rate in the underlying network.

The cloud framework is used to train the model for dynamic anomaly signature detection of the incoming traffic. The trained model is implemented on the Filter Agent configured on the underlying network for continuous packet analysis (Steps 15-18).

Algorithm 2 Adaptive SVM-based Anomaly Detection

```
INPUT: Incoming Traffic: $\mathbb{T}$
OUTPUT: IIoT devices ranking: $\mathbb{R}$
1:  while ($\mathbb{T} \neq$ null) do
2:      Packet Analyzer: ($\mathbb{H}$,$\mathbb{P}$) $\leftarrow \mathbb{T}$        ▷ $\mathbb{H}$ : Header, $\mathbb{P}$ : Payload
3:      Forward: $\mathbb{H} \rightarrow$ SVM
4:      Forward: $\mathbb{P} \rightarrow$ Cloud
5:      Classify $\mathbb{H}$                                   ▷ Refer Sec. IV
6:      if Label==-1 then                                ▷ -1: Anomalous
7:          Generate: Packet_In
8:          SDN Create Flow Rule: $\mathbb{F}_{rule}$
9:          Packet_Out($\mathbb{F}_{rule}$) $\rightarrow$ Switches
10:         Update: $\mathbb{R}$                                ▷ Ranking
11:     else
12:         Fetch: $\mathbb{F}_{rule} \leftarrow$ Flow Table
13:         Redirect: $\mathbb{T}$ as per $\mathbb{F}_{rule}$
14:     end if
15:     Fetch: $\mathbb{P}$
16:     Forward: $\mathbb{P} \rightarrow \mathbb{V}_m$                   ▷ $\mathbb{V}_m$ : Virtual Machine
17:     Training SVM on $\mathbb{V}_m$                         ▷ Refer Sec. IV
18:     Implement: $\mathbb{T}_m \rightarrow$ Filter Agent          ▷ $\mathbb{T}_m$: Trained Model
19: end while
```

V. Result and Discussion

The considered dataset and the evaluated results simulating the proposed scheme are discussed in the subsequent sections.

A. Datasets Used

Network-based Dataset. The statistical information about the network is gathered in the dataset[3] and further used to train

[3] https://research.unsw.edu.au/projects/unsw-nb15-dataset

and test the model. There are 49 classes labeled network-based features in the included dataset. The data pre-processing of all the network features in the dataset considering the mean, standard deviation, minimum, and maximum range of the numbers is shown in Table II. The feature, *attack_cat*, indicates the type of the synthetic attack generated in the network.

Environments. For the proposed scheme's experimental testbed, a Spark YARN cluster of 1 master and 3 workers are configured, each with a capacity of 4 cores and 16 GB of memory. The operating system for each node is Ubuntu Server 20.04 LTS, SSD Volume Type. The Spark version is 3.3.0, and the Hadoop version is 3.3.4.

TABLE II: Statistics of the Dataset

Features	**Mean**	**Std. dev.**	**Min.**	**Max.**
dur	1.359389	6.480249	0	59.99999
proto	109.6067	22.3526	0	132
service	1.61892	2.305151	0	12
state	2.355176	0.8679419	0	8
spkts	20.29866	136.8876	1	9616
dpkts	18.96959	110.2583	0	10974
sbytes	8844.844	174765.6	28	12965230
dbytes	14928.92	143654.2	0	14655550
rate	95406.19	165401	0	1000000
sttl	179.547	102.94	0	255
dttl	79.60957	110.5069	0	254
sload	73454030	188357400	0	5988000000
dload	671205.6	2421312	0	22422730
sloss	4.953	66.00506	0	4803
dloss	6.94801	52.733	0	5484
sinpkt	985.9769	7242.246	0	84371.5
dinpkt	88.2163	987.0932	0	56716.82
sjit	4976.254	44965.85	0	1460480
djit	604.3538	4061.043	0	289388.3
swin	116.2573	127.001	0	255
stcpb	969250400	1355264000	0	4294959000
dtcpb	968877000	1354000000	0	4294882000
dwin	115.0136	126.8865	0	255
tcprtt	0.04139564	0.07935397	0	2.518893
synack	0.02102045	0.04339978	0	2.100352
ackdat	0.02037519	0.04050636	0	1.520884
smean	136.7518	204.6774	28	1504
dmean	124.1734	258.3171	0	1458
trans_depth	0.1059821	0.7769108	0	172
response_body_len	2144.292	54207.97	0	6558056
ct_srv_src	9.306437	10.70433	1	63
ct_state_ttl	1.304179	0.9544061	0	6
ct_dst_ltm	6.193936	8.052476	1	51
ct_src_dport_ltm	5.383538	8.047104	1	51
ct_dst_sport_ltm	4.206255	5.783585	1	46
ct_dst_src_ltm	8.729881	10.95619	1	65
is_ftp_login	0.01494802	0.126048	0	4
ct_ftp_cmd	0.01494802	0.126048	0	4
ct_flw_http_mthd	0.1330664	0.7012076	0	30
ct_src_ltm	6.955789	8.321493	1	60
ct_srv_dst	9.100758	10.75695	1	62
is_sm_ips_ports	0.01575216	0.1245155	0	1
Feature	0.8196999	0.3844383	0	1
Average	**50616.73**	**87671**	**1.0**	**17534.1**

B. Results

This section discusses the performance of binary tree-based flow table optimization, energy consumption, tactile/non-tactile-based traffic classification and anomaly detection.

1) Flow Table Optimisation: The proposed scheme aims to achieve the tactile characteristics in the underlying network. The performance of the proposed model is calculated considering two scenarios: a) the performance metrics using the SOM model, and b) the performance metrics related to data insertion time, latency, and lookup time to search for flow entry in the flow table. In the binary tree-based table optimization approach, the incoming traffic is segregated into two classes, namely tactile and non-tactile-based applications. The categorization of the network applications into tactile and non-tactile applications is presented in Table I. Time-sensitive and real-time processing-based applications are classified as tactile-based applications, and others are labeled as non-tactile-based applications.

Considering the above mentioned scheme, the flow rules are segregated into two flow tables: Tactile-based and Non-tactile-based. In this context, Fig. 6 shows the results related to the insertion time of flow entries, comparing the binary tree-based approach with the standard/baseline approach.

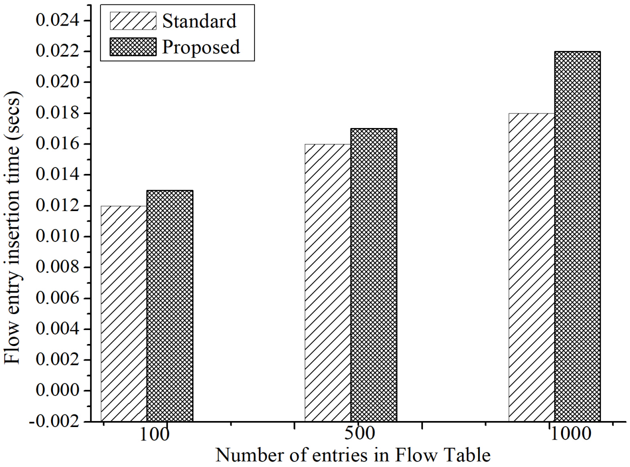


Fig. 6: Binary Table Insertion Overhead

Fig. 7 demonstrates that the lookup time for proposed scheme outperforms the standard approach, thus validating the conditions of tactile network. It shows that the proposed approach handles and processes application priorities effectively.

2) Energy Consumption: Classifying traffic into tactile and non-tactile categories, rather than treating all traffic as tactile in IIoT environments can save a substantial amount of energy. Differentiating between tactile and non-tactile traffic processing requirements and corresponding resource demands leads to this efficiency. High-power setups are frequently used in tactile applications, which demand minimal latency, better responsiveness and real-time processing. However, non-tactile traffic can be handled with lower-performance setups or less power-intensive equipment because it may tolerate some delay. By dividing traffic and assigning resources according to their requirements, the system can prevent over-provisioning and lower total power consumption. The energy consumption to process the specified entries in the flow table is compared in the Fig. 8. The numbers clearly demonstrate that the proposed approach is energy-efficient when matching rules from both tactile and non-tactile-based flow tables. Binary-based flow tables reduce memory utilization and enhance lookup speed,

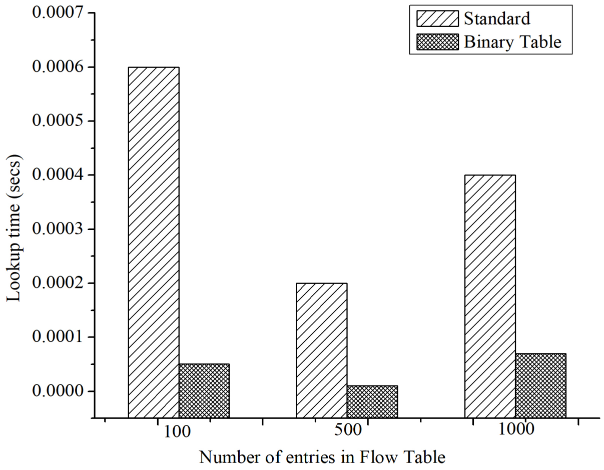


Fig. 7: Binary Table Lookup Time Overhead

leading to a direct reduction in energy consumption during network operations.

In similar manner, the Fig. 9 reflects the energy consumption while processing the classified traffic as compared to Non-classified traffic in IIoT environment.

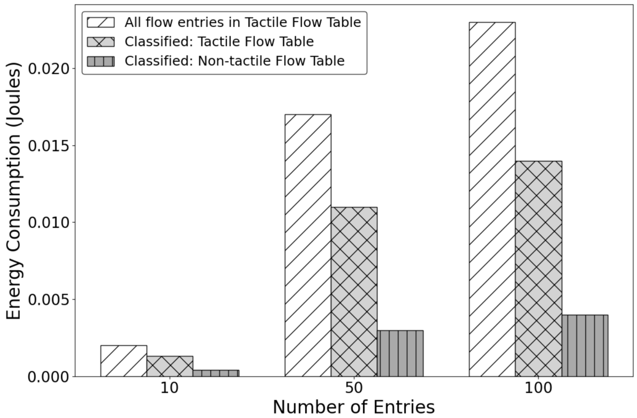


Fig. 8: Energy Consumption: Processing Flow Entries

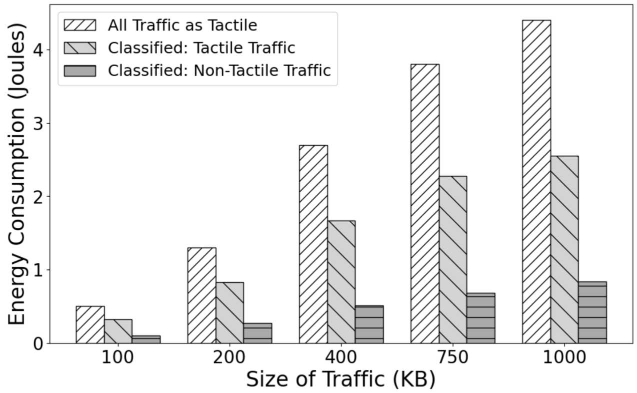


Fig. 9: Energy Consumption: Processing Incoming Traffic

3) Anomaly Classification: SOM is used to classify the incoming traffic and accordingly label it as Tactile-based Services or Non-Tactile-based Services Traffic mentioned in Table I. The network traffic metrics are analysed using the SOM classification approach, and Fig. 10 shows the SOM-based distance map over the number of collected metrics.

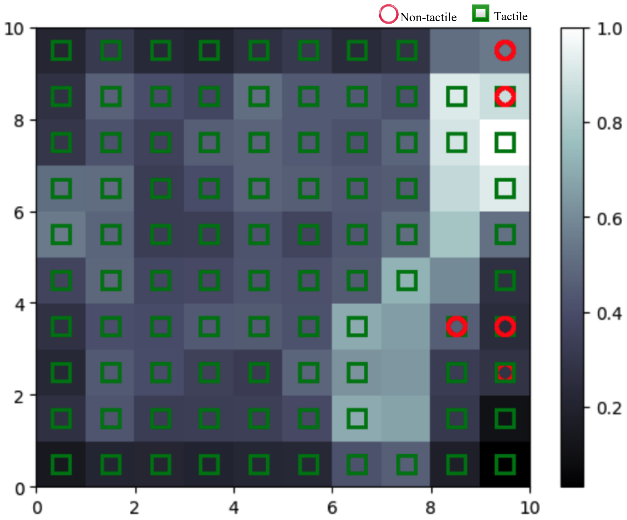


Fig. 10: SOM-based Tactile and Non-Tactile Application Classification

In Fig. 10, the red maps are the metrics classified as non-tactile-based applications, and the maps in green are the tactile-based applications. The performance of SOM-based classification is highlighted in the Fig. 11. The proportion of correctly classified data points using defined dataset (Accuracy) is calculated by using the following metrics.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

where, TP is the True Positive, FP is the False Positive,FN is the False Negative,and TN is the True Negative labels. The Precision is calculated by using the below mentioned equation:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{6}$$

The true positive rate (Recall) is calculated by using the below mentioned equation:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{7}$$

The F1 Score is calculate using Eq. 8.

$$\text{F1 Score} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{8}$$

4) Anomaly Detection: The underlying network is configured with a trained model (SVM) for detecting anomalous traffic and stopping positive traffic (Anomalous) for further processing. The simulated results of training and testing the SVM model is promising and highlighted in Fig. 12 with the numbers to label the traffic as normal or anomalous.

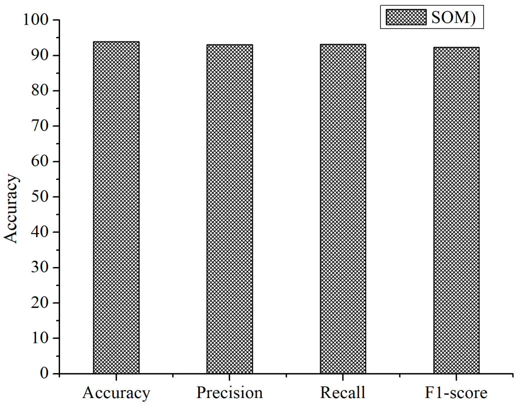


Fig. 11: Accuracy of SOM Model

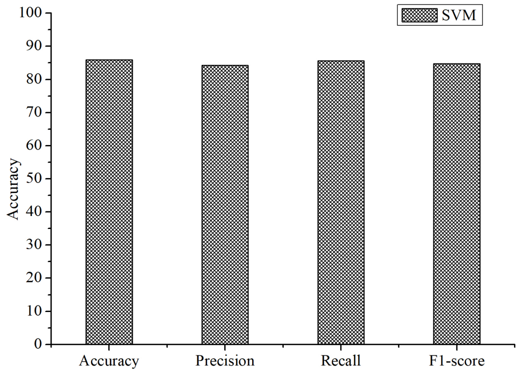


Fig. 12: Accuracy of SVM Model

VI. Conclusion

The rapid development of IIoT paradigm has led to a new era of enhanced monitoring and control of industrial applications. The timely transfer of industrial data and handling anomalies in data traffic is essential in IIoT environment, as otherwise it may impact overall operational performance. Complex environments and limited communication bandwidth make it difficult for underlying network to process data traffic with tactile guarantees. This paper proposes an energy-efficient tactile data sharing mechanism that differentiates the tactile traffic from the normal one and process it with lower lookup time using a dynamic SDN policy. Additionally, this work adopts SVM to detect anomalies that can hinder the performance of industrial data processing. The outcomes look promising in terms of reduced overheads, lookup time and energy consumption while achieving adequate accuracy for the SOM and SVM models. Future work includes using edge and cloud registering to decrease latency and improve energy effectiveness in tactile-driven frameworks. Deploying anomaly detection algorithms on edge devices facilitates real-time analysis with minimal latency and enhanced scalability.

Acknowledgement

This work is supported by the National Natural Science Foundation of China under Grant 62402257 and 62471493, the Natural Science Foundation of Shandong Province under Grant ZR2023LZH017, ZR2022LZH015, and ZR2024MF066, the Open Foundation of Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Qilu University of Technology (Shandong Academy of Sciences) under Grant 2023ZD010, the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) under Grant RS-2024-00392332, the National Research Foundation of Korea under Grant NRF-24109, and the China University Research Innovation Fund under Grant No.2023IT207.

References

[1] M. Aazam, S. Zeadally, and K. A. Harras, "Deploying fog computing in industrial internet of things and industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674–4682, 2018.

[2] B. Ahlgren, M. Hidell, and E. C.-H. Ngai, "Internet of things for smart cities: Interoperability and open data," *IEEE Internet Computing*, vol. 20, no. 6, pp. 52–56, 2016.

[3] L. Chettri and R. Bera, "A comprehensive survey on internet of things (iot) toward 5g wireless systems," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16–32, 2020.

[4] S. Kar, P. Mishra, and K.-C. Wang, "5g-iot architecture for next generation smart systems," in *5GWF'21*, 2021, pp. 241–246.

[5] K. S. Kim, D. K. Kim, C.-B. Chae, S. Choi, Y.-C. Ko, J. Kim, Y.-G. Lim, M. Yang, S. Kim, B. Lim, K. Lee, and K. L. Ryu, "Ultrareliable and low-latency communication techniques for tactile internet services," *Proceedings of the IEEE*, vol. 107, no. 2, pp. 376–393, 2019.

[6] S. Bera, H. Das, S. Nayak, and R. Patgiri, "Future tactile internet: Issues, challenges and applications," in *ISPCC'21*, 2021, pp. 625–630.

[7] M. Y. Arafat and S. Moh, "Routing protocols for unmanned aerial vehicle networks: A survey," *IEEE access*, vol. 7, pp. 99 694–99 720, 2019.

[8] J. Wagner, H. Morath, J. Zhang, F. Wieczorek, L. Lüneburg, F. H. P. Fitzek, and G. T. Nguyen, "Tactile electronics meets softwarised networks," in *CCNC'22*, 2022, pp. 961–962.

[9] R. Amin, M. Reisslein, and N. Shah, "Hybrid sdn networks: A survey of existing approaches," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3259–3306, 2018.

[10] R. Chaudhary, G. S. Aujla, N. Kumar, and P. K. Chouhan, "A comprehensive survey on software-defined networking for smart communities," *International Journal of Communication Systems*, p. e5296, 2022.

[11] G. Han, J. Tu, L. Liu, M. Martínez-García, and Y. Peng, "Anomaly detection based on multidimensional data processing for protecting vital devices in 6g-enabled massive iiot," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5219–5229, 2021.

[12] W. Wen, U. Demirbaga, A. Singh, A. Jindal, R. S. Batth, P. Zhang, and G. S. Aujla, "Health monitoring and diagnosis for geo-distributed edge ecosystem in smart city," *IEEE Internet of Things Journal*, 2023.

[13] S. Sciancalepore, O. A. Ibrahim, G. Oligeri, and R. Di Pietro, "Pinch: An effective, efficient, and robust solution to drone detection via network traffic analysis," *Computer Networks*, vol. 168, p. 107044, 2020.

[14] H. Nizam, S. Zafar, Z. Lv, F. Wang, and X. Hu, "Real-time deep anomaly detection framework for multivariate time-series data in industrial iot," *IEEE Sensors Journal*, vol. 22, no. 23, pp. 22 836–22 849, 2022.

[15] U. Demirbaga, G. S. Aujla, M. Singh, A. Singh, H. Sun, and J. Camp, "An intelligent monitoring and warning framework in drone swarm digital twin systems," in *ICC'24*, 2024, pp. 1945–1950.

[16] H. Algamdi, G. S. Aujla, A. Jindal, and A. Trehan, "Intrusion detection in critical sd-iot ecosystem," in *ICC'23*, 2023, pp. 1559–1564.

[17] R. Sallé de Chou, M. A. Srir, L. Najman, N. Passat, H. Talbot, and I. Vignon-Clementel, "Convex optimization for binary tree-based transport networks," in *International Conference on Discrete Geometry and Mathematical Morphology*. Springer, 2024, pp. 217–228.

[18] "The tactile internet," *ITU-T Technology Watch Report*, 2014, [Accessed Jul 2023]. [Online]. Available: https://www.itu.int/dms_pub/itu-t/oth/23/01/T23010000230001PDFE.pdf

[19] P. Russell, M. A. Elsayed, B. Nandy, N. Seddigh, and N. Zincir-Heywood, "On the fence: Anomaly detection in iot networks," in *NOMS'23*, 2023, pp. 1–4.