2025 3rd International Conference on Intelligent Systems, Advanced Computing and Communication (ISACC) | 979-8-3315-2389-3/25/$31.00 ©2025 IEEE | DOI: 10.1109/ISACC65211.2025.10969354

Tiny ML-based Secure and Energy Efficient Unmanned Aerial Vehicles Surveillance Framework for Smart Cities

Aditya Shah
Dept. of Comp. Science & Engg.
Institute of Tech., Nirma University
Ahmedabad, India
21bce008@nirmauni.ac.in

Vanera Vivek
Dept. of Comp. Science & Engg.
Institute of Tech., Nirma University
Ahmedabad, India
21bce313@nirmauni.ac.in

Dishant Savaliya
Dept. of Comp. Science & Engg.
Institute of Tech., Nirma University
Ahmedabad, India
21bce261@nirmauni.ac.in

Rajesh Gupta
Dept. of Comp. Science & Engg.
Institute of Tech., Nirma University
Ahmedabad, India
rajesh.gupta@nirmauni.ac.in

Sudeep Tanwar
Dept. of Comp. Science & Engg.
Institute of Tech., Nirma University
Ahmedabad, India
sudeep.tanwar@nirmauni.ac.in

Jitendra Bhatia
Dept. of Comp. Science & Engg.
Institute of Tech., Nirma University
Ahmedabad, India
jitendra.bhatia@nirmauni.ac.in

***Abstract*—Recent advances in smart cities require a more effective, secure, and energy-efficient surveillance system to protect people. This paper proposes a multi-layered architecture for a UAV-based traffic monitoring system, designed to enhance situational awareness and system resilience in real-time traffic environments. The architecture consists of four key layers: the Traffic Layer, where the UAV continuously monitors and gathers data from diverse traffic entities; the UAV Layer, responsible for data collection and packet transmission; the Security Layer, which employs quantized TinyML models, including decision trees, artificial neural networks (ANN), and support vector machines (SVM), to detect and mitigate Denial-of-Service (DoS) attacks; and the Controller Layer, which manages verified data and orchestrates system actions based on traffic conditions and cyber security alerts. Implementing machine learning algorithms optimized for resource efficiency using quantization techniques ensures secure communication, real-time data processing, and energy conservation within the UAV. Our results demonstrate the system's ability to maintain robust performance under complex traffic conditions while effectively mitigating potential cyber security threats in an energy efficient manner.**

***Index Terms*—UAV, ANN, SVM, Decision tree, TinyML, Artificial Intelligence, DoS**

I. Introduction

The increase in the complexity of the *smart cities* in the coming years requires the development of energy-efficient surveillance techniques in public areas. The concept of *Unmanned Aerial Vehicles (UAVs)* [1] has been widely accepted because it offers a relatively flexible and mobile platform for real-time surveillance. Most UAVs are equipped with sensors, cameras, and communication modules, which enable these systems to acquire, process, and transmit valuable information for various urban monitoring purposes. This is because they are capable of covering large areas of interest; they can monitor events from the sky; and they offer clear images for analysis. However, despite the advantages it brings, UAVs are subject to several security risks, which include the ability to penetrate UAV control systems, data theft, and vulnerability to cyber terrorism. Therefore, there is a growing effort to improve the security of UAV systems through encryption and communication security and Machine learning algorithms for the detection of threats.

Although some advancements have been made in this field, *Tiny Machine Learning (TinyML)* [2] offers the possibility to enhance UAV systems even more. TinyML allows the ML models to be implemented on IoT and other devices with low compute resources and limited battery capabilities which is the case with UAVs as well. There is one major challenge that UAVs experience in the present day, and that is the utilization of *energy* which reduces their flight duration and effectiveness. In traditional approaches, much of the computation load is shifted to servers in the cloud, which increases latency and contributes to security concerns. Through the concepts of *on-device data processing* and *real-time decision-making*, TinyML minimizes the amount of time spent to relay data to distant servers for interpretation, enhancing both *energy consumption and data privacy* [3]. While researchers have presented numerous lightweight algorithms as well as security protocols, there is still a requirement for *progression* in the implementation of TinyML into UAV systems for enhancing energy and security constraints [4].

Inspired by such continuous threats and the prospects of TinyML, we proposed a *secure and energy-efficient unmanned aerial vehicle (UAV) surveillance system* for smart cities. Our framework incorporates TinyML models learned on a particular dataset that we have collected for private use in urban surveillance tasks, for instance, object recognition, traffic control, and surveillance of security risks.

In the process of designing the smart city environment, we took into account several realistic implementations of the smart city and hence coordinated our design to make the system more energy efficient as well as bringing in a higher level of security for UAV operations. This solution offers real-time monitoring and threat identification for the UAVs, while at the same time enabling the UAVs to stay airborne for longer durations with no threat to their security.

A. Research Contribution

The contributions of this paper are as follows:

- **TinyML-Based UAV Surveillance Framework**: We propose a TinyML-based framework that enables UAVs to perform energy-efficient real-time surveillance with very little dependence on external resources.
- **Security Enhancements**: We address data security concerns by implementing on-device machine learning, which significantly reduces the risk of cyber-attacks and data breaches during surveillance operations.
- **Energy Effective Model Conversion**: To handle the energy constraint for our UAV, we utilized various Python libraries to convert our ML models into Tiny ML formats thereby reducing the load on the UAVs.
- **Custom Dataset and Model Training**: A private dataset was collected to train our TinyML models, ensuring that our framework is optimized for real-world surveillance tasks, such as anomaly detection in packet transmission.

B. Paper Organization

The remainder of the paper is structured as follows:

- **Section 2** introduces the *proposed architecture* of the TinyML-based UAV surveillance system, including its design and key components.
- **Section 3** describes the *dataset*, detailing the features utilized for the training of our models.
- **Section 4** presents the *performance evaluation*, highlighting the energy consumption, security measures, and accuracy of the proposed framework.
- **Section 5** concludes the paper by summarizing the findings and providing potential directions for future work.

II. Proposed Architecture

Fig. 1 presents the proposed TinyML-based architecture for UAV surveilliance framework for smart cities.

A. Traffic Layer

The traffic layer in this architecture is a dynamic layer that changes from time to time through constant monitoring of the UAV's surrounding traffic activity. It includes road users like pedestrians, automobiles like cars, buses, trucks, and motorcycles together with movable and stationary objects on the road. This layer is designed to simulate real-world traffic scenarios, where the UAV has to adapt to the speed, motion, unpredictability, etc. The primary aim of the UAV in this layer is to acquire adequate traffic information and threatened points for assessing the general situation of the system. This layer also considers other possible cyber security threats, where the attackers want to attempt a Denial-of-Service (DoS) attack on the UAV to interfere with its surveillance capabilities. These attacks could interfere with normal communication systems, affect the decision-making and control mechanisms, or temporarily disable the UAV's ability to monitor traffic effectively. The UAV must be able to handle these threats while ensuring continuous surveillance and traffic monitoring in an energy-efficient manner without sending the signals to a remote server which would cause latency issues.

B. UAV Layer

In the UAV layer, the role of the UAV is to capture data in real-time required by the environment from its surroundings [6]. Traffic, cars, pedestrians, obstacles, and other essential entities are recognized by the UAV. It captures data from the sensors such as the distance to the object, their velocity and trajectory, and any danger that may affect the environment. This data is converted into packets and sent to the security layer for validation and analysis.

Each packet that is generated by the UAV contains several layers that are useful in communication as well as in processing. The packet begins with Packet ID, which helps the security layer to monitor and control each packet going through the system. The source and destination addresses allow the packet to reach its right destination in the network while the packet size defines the overall amount of data being transmitted, and helps to manage bandwidth and other transmission resources.

Moreover, the packet has a Payload section, which is actual environment information. It involves raw data gathered by the UAV's onboard systems, like LiDAR, camera, and GPS, and gives information on object distances, speeds, and trajectories. UAV status information such as battery status, signal quality, and sensor conditions are also embedded in the packet to measure system health.

After these packets are formed, they are forwarded to the security layer [7], where the TinyML algorithms are applied. These algorithms are crucial to differentiating normal packets from possibly malicious DoS attacks. The security layer used Machine Learning models to examine packet characteristics of the packets for frequency, size distribution, and any abnormal data that may point to the signs of a threat. By this process, the packets are differentiated into two classes namely benign and DoS, and sent to the Controller Layer for further application. Therefore making the system safe and operational in most cases of complexity. The UAV layer plays a crucial role in real-time data collection and secure communication in the network architecture.

C. Security Layer

The primary focus of the security layer is to ensure the security and integrity of data packets received from UAVs. This layer would be responsible for processing

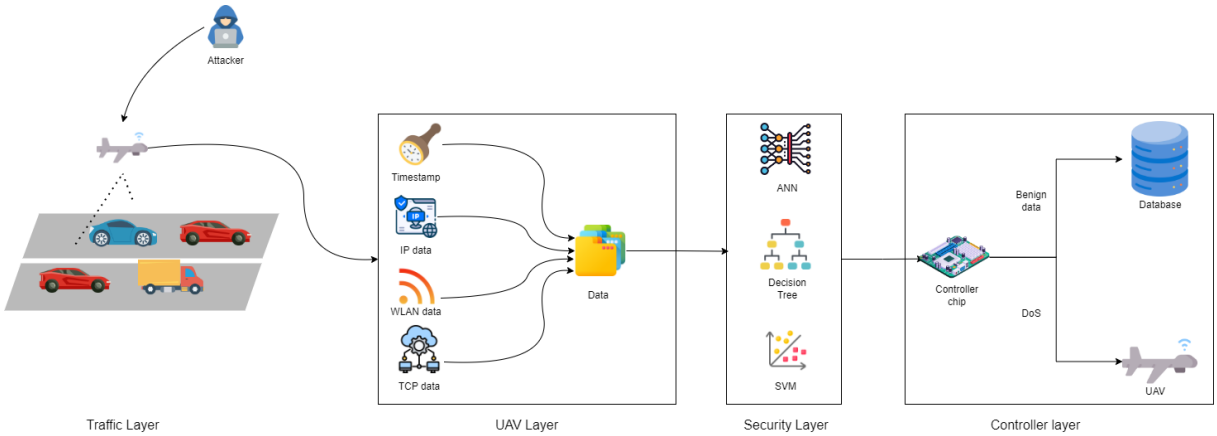

Fig. 1: Proposed TinyML Architecture [5].

incoming data packets for security concerns using various TinyML models, which are lightweight versions of machine learning algorithms. These models have reduced size and complexity which is achieved via quantization. In quantization, parameters of the model expressed in high-precision numbers are mapped to smaller precision numbers requiring less memory and computation. This allows models to run efficiently consuming less power and increasing operate ring time.

The TinyML models in the Security Layer are responsible for detecting any attempt at malicious activities, such as Denial-of-Service(DoS) attacks. These models are integrated within the UAV, enabling real-time packet analysis and decision-making at the edge. When the UAV transmits data packets containing various information about data packets like packet length, TCP protocols, IP address, packet type, time taken for transmission, etc. When the UAV receives these packets, using this information the models analyze the nature of transmission and any possible threats.

If the packet is recognized as benign then it is forwarded to a database for storage and then used by the Controller Layer. However, if packets show characteristics of any possible security invasion, then it immediately triggers a warning to the controller layer to take swift action against such potential threats.

By quantization of models, the processing time and energy load on UAVs is reduced significantly with real-time security analysis. This extends the operational efficiency of the model and reduces reliance on external systems for threat detection traffic data from vehicles.

1) Models Used in the Security Layer: The Dataset was utilized to train various classification models in the security layer to classify the incoming packets from the UAV as either Benign or Denial of Service packets. After training, using a Tiny ML-based library, these models were converted to a tiny format. By converting these models their size and computer time decreased. These are a few models utilized in this experiment :

- **Decision Tree**: Decision Tree is a machine learning model that is non-parametric and supervised in nature. This model is utilized for both classification and regression tasks. In our experimentation, this model was trained to classify whether a given packet is benign or malicious. This was based on features of data packets mentioned in dataset.[8]. The tree structure is built by breaking down decisions into simpler conditions based on features. This approach of modeling provides a transparent and interpretable model for understanding the reasoning behind decision-making.
- **Artificial Neural Network (ANN)**: An Artificial Neural Network (ANN) is a deep learning model that mimics the human brain's neural structure. For our application of cyber-security [9], an ANN was trained to classify the incoming packets by learning complex patterns in the packet data.[10] The ANN model utilized had 7 layers. Quantization of the ANN has been applied to reduce the size of weights and activations, allowing for faster inference and lesser memory requirement on resource-constrained devices.

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 1000)	38,000
dense_1 (Dense)	(None, 500)	500,500
dense_2 (Dense)	(None, 200)	100,200
dense_3 (Dense)	(None, 100)	20,100
dense_4 (Dense)	(None, 50)	5,050
dense_5 (Dense)	(None, 10)	510
dense_6 (Dense)	(None, 1)	11

Total params: 664,371 (2.53 MB)
Trainable params: 664,371 (2.53 MB)
Non-trainable params: 0 (0.00 B)

Fig. 2: ANN model architecture

Figure 2 shows the architecture of the ANN model describing the number of neurons in each layer and the activation function utilized. After training the model for 10 epochs, a loss curve was utilized for observing model performance (as shown in Fig. 3). The loss curve suggested that the model improved performance over time as both training and validation loss decreased. During the last few epochs, the loss curve converged

to a steady value indicating to point of stabilization in predictions.

Fig. 3: ANN Model Loss Curve

- **Support Vector Machine (SVM)**:
  Based on the dataset utilized by us for the classification problem, Support Vector Machines (SVM) [11] proved to be a useful option as well. It is a supervised learning algorithm that finds the most optimal hyperplane required to classify the data points into two different categories. In our use case, the two different categories are benign packets and DoS packets. Thus, SVM's aim would be maximizing the margin between these packets and making it an effective algorithm for classification with a well-defined boundary.

The models are then reduced to Tiny Format by using the ONNX and TensorFlow Lite libraries and observed based on set criteria like inference time, memory difference, and inference accuracy.

2) Quantization and TinyML: The models utilized in this experiment, namely ANN, SVM, and decision tree have been all converted to Tiny Formats through the usage of quantization [6] methods to improve the inference time and memory usage of each model. The process of quantization converts high-precision models into a form that uses fewer bits without a significant reduction in the model accuracy. Typically, the 32-bit floating points are converted into 8-bits. This process allows the UAV to locally run Machine Learning models, making it capture real-time threats in an energy-effective manner.

D. Controller Layer

The controller layer works as a decision-making layer and it is responsible for processing verified data packets and managing system actions. Algorithm 1 shows the step-by-step process of packet processing. When the controller layer receives benign packets from the security layer, the controller layer analyzes the traffic data, including system health metrics and hazard reports, as well as makes decisions like controlling UAVs or changing the traffic path. It also controls the information storage, which means that the benign data is properly stored for future analysis, system optimization, or training new machine learning models. In cases of abnormal behavior, like a DoS attack, the controller layer rapidly responds by applying countermeasures including quarantine or switch channels, and hence maintains the robustness of the system.

Algorithm 1 Packet processing algorithm in the controller layer.

Input: Packet P from Security Layer
Output: Appropriate action based on packet type
ProcessPacket(P):
if P is DoS Attack **then**
    Counteract Attack
    Apply countermeasures *Switch Communication Channel* or *Quarantine Affected Nodes*
else if P is Benign **then**
    Store P in *Database*
end if

III. Dataset

The dataset contains various features capturing details of network traffic. In general, these features include:

- **Frame Information**: Includes attributes such as frame content, frame number, frame length, frame protocols, and WLAN-specific details.
- **IP Information**: Consists of IP addresses for source and destination.
- **Network Protocol Information**: Includes TCP source and destination ports, TCP sequence, and acknowledgment numbers.
- **Frame Control Information**: Includes frame type and subtype details.
- **Timing Information**: Time since the last packet was received.
- **Target Variable**: Indicates whether the network traffic is benign or DoS.

Many other features describe various other aspects of the data frames captured and these are utilized further in model training.

IV. Performance Evaluation

A. Experimental Setup

In this study, we utilized Google Colab, a cloud-based platform equipped with GPUs, to leverage its computational power and the latest software versions for model development and evaluation. The environment was configured with Python 3.9 and TensorFlow 2.12, which facilitated efficient model training and deployment. Essential libraries included Pandas for data manipulation and preprocessing, enabling robust data cleaning and transformation; NumPy for performing numerical operations; and Scikit-Learn for implementing and evaluating traditional machine learning models such as Decision Trees and Support Vector Machines (SVMs). TensorFlow was employed to construct and train a complex artificial neural network (ANN), which was later converted to TensorFlow Lite format for TinyML applications using quantization techniques to optimize for mobile and edge devices. The ONNX library, along with ONNX Runtime,

was utilized to convert and evaluate the models in the ONNX format, allowing for comparisons between different machine learning frameworks. Additionally, joblib was used for saving and loading models, while Matplotlib and Seaborn were instrumental in visualizing performance metrics, including model accuracy, size, and inference time. This combination of tools provided a comprehensive framework for evaluating and comparing the efficiency and performance of various models across different formats and environments.

B. Size Difference

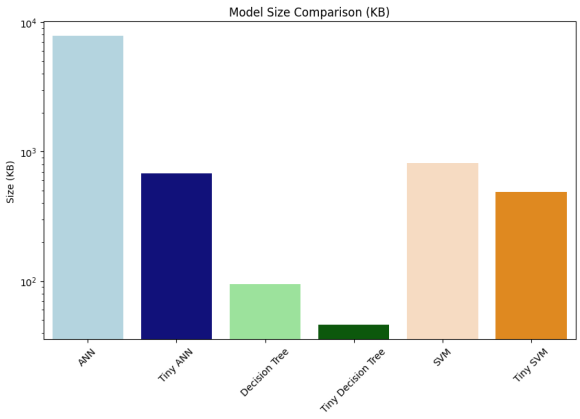

Fig. 4: Memory Size of Different Machine Learning Models

Figure 4 shows the difference in the size of the three different machine learning models namely ANN, Decision Tree and SVM.

- ANN: It is evident that there is a significant decrease in the size of our ANN model. The size of the ANN model initially was 7792.38 KB. After utilizing Tiny ML and converting it to the TensorFlow Lite Model using Quantization, the size of the model has been reduced to 677.17 KB. There was around 91 percent reduction in the size of our ANN model which would prove to be significantly efficient and energy-conserving for our UAV.
- Decision Tree: For the decision Tree model that we implemented on our dataset for UAV detection, the initial size was 94.32 KB. After utilizing the ONNX library to convert it into a tiny version of the Decision Tree which was 45.82 KB. This reduction is around 52 percent. While this may not be as significant as ANN, it is still an extremely beneficial reduction of load for our UAV.
- SVM: In SVM, initially the size of the model was 817.29 KB. The tiny version of it created by using the ONNX library was of size 488.92 KB. The reduction in the size of the SVM model is 40 percent. This reduction was the least as compared to other models but in no way can it be considered insignificant.

From the results observed in the sizes of these models, we saw that Artificial Neural Networks have the most benefit in conversion to Tiny size. While the models used by us were relatively smaller in size, the percentage reduction in their size was visibly evident. As the size of our model would increase for different applications, reducing its size using various quantization methods would become a necessity to reduce power load and consumption.

C. Inference Time

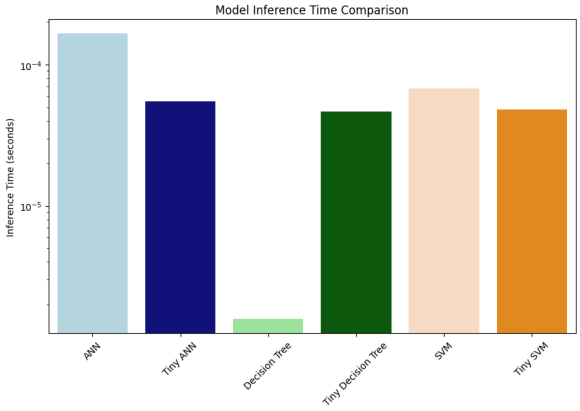

Fig. 5: Memory Size of Different Machine Learning Models

Figure 5 shows the difference in inference time of the three different machine learning models namely ANN, Decision Tree, and SVM.

- ANN: In the ANN, the inference time did reduce significantly. The standard ANN took 0.000166 seconds per sample whereas tiny ml took 0.000055 seconds per sample which is 33 percent of the inference time of the standard-sized model, which suggests that tiny ANN models have faster inference time.
- Decision Tree: In Decision Tree models, the standard Decision Tree model had an inference time of 0.000002 seconds per sample while the tiny Decision Tree model had an inference time of 0.000046 seconds per sample which is significantly higher. This suggests that Tiny ML conversion for decision trees may not be a viable approach.
- SVM: In the SVM machine learning model, the standard-sized model had an inference time of 0.000068 seconds per sample and the tiny SVM model had an inference time of 0.000048 seconds per sample. A 30 percent reduction in inference time was observed.

This experimentation in inference suggests that tiny models except for the Decision tree have an average inference time of tiny models that is 30 percent less than standard models. This reduction in inference time suggests better utilization of computing resources and would be useful in the longer operations of UAVs.

D. Accuracy

Figure 6 shows the difference in the accuracy of the three different machine learning models namely ANN, Decision Tree, and SVM.

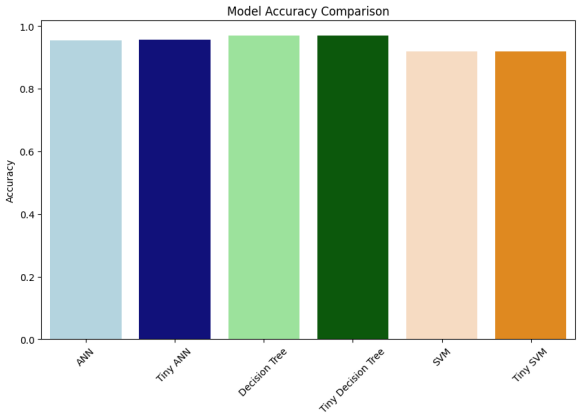

Fig. 6: Accuracy of Different Machine Learning Models

- ANN: In the ANN, the accuracy of the standard model was observed to be 0.9549 and of the Tiny model to be 0.9552. The accuracy of the Tiny model is better compared to the standard model however difference observed is very less.
- Decision Tree: In Decision Tree models, the standard Decision Tree model had an accuracy of 0.9684 and of the tiny model was 0.9684. No difference was observed between the accuracy values of these models.
- SVM: In the SVM machine learning model, the standard-sized model had an accuracy of 0.9182 and the tiny model was 0.9182. Similar to Decision Tree models no differences in accuracy values were observed here.

This experimentation in the accuracy of the tiny model and the standard model shows that there is no compromise in accuracy despite a significant reduction in size and inference time which is very important for UAVs for performing their primary task. This experiment also shows that the decision tree model had better accuracy compared to ANN and SVM despite lesser inference time and model size.

V. UAV Surveillance Frameworks and Trade-Offs

- Several UAV surveillance frameworks have been developed but they often struggle with energy efficiency and security challenges, limiting their effectiveness in real-world applications.
- Traditional cloud-based systems rely on remote servers for data processing causing high latency, increased energy consumption and security risks associated with data transmission over the network.
- UAVs with basic onboard processing have limited analytical capacities struggling with complex data interpretation along with increased energy usage.
- Swarm intelligence enhance surveillance through coordinated UAV operations, but lack robust security protocols, making them susceptible to cyber threats.
- Some UAV systems integrate machine learning models but these are not optimized for low-power devices, these models require high computational resources and high latency.

VI. Conclusion

Our study demonstrates that Tiny ML-based models can be utilized for secure and energy-efficient UAV surveillance frameworks in smart cities. By converting standard models to tiny formats, we were able to significantly reduce the size and inference time of models. This reduction in model size and inference time can significantly increase the operating time and energy efficiency in UAVs. While standard models showed good accuracy of more than 90 percent, the tiny ml-based model had comparable accuracy, reducing computational resources and power consumption suited for resource-constrained environments.

In future works, optimization in the integration of tiny ml-based models can be studied along with more features to detect cyber threats by these models can be explored. The complexity of the models used can be increased by improving deep-learning models like Transformers, CNNs and LSTMs. The results of this study conclude the viability of Tiny ML-based models for UAVs in smart cities for better operations.

References

[1] A. Giyenko and Y. I. Cho, "Intelligent uav in smart cities using iot," in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, pp. 207–210, 2016.

[2] H. Han and J. Siebert, "Tinyml: A systematic review and synthesis of existing research," in *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 269–274, 2022.

[3] A. Dutta and S. Kant, "Implementation of cyber threat intelligence platform on internet of things (iot) using tinyml approach for deceiving cyber invasion," in *2021 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME)*, pp. 1–6, 2021.

[4] R. Immonen and T. Hämäläinen, "Tiny machine learning for resource-constrained microcontrollers," *Journal of Sensors*, vol. 2022, pp. 1–11, 11 2022.

[5] Flaticon, "https://www.flaticon.com/." http://aiweb.techfak.uni-bielefeld.de/content/bworld-robot-control-software/, 2008. [Online; accessed 19-July-2008].

[6] Z. Wei, M. Zhu, N. Zhang, L. Wang, Y. Zou, Z. Meng, H. Wu, and Z. Feng, "Uav-assisted data collection for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15460–15483, 2022.

[7] C. Yin, Z. Xiao, X. Cao, X. Xi, P. Yang, and D. Wu, "Enhanced routing protocol for fast flying uav network," in *2016 IEEE International Conference on Communication Systems (ICCS)*, pp. 1–6, 2016.

[8] S. Sahu and B. M. Mehtre, "Network intrusion detection system using j48 decision tree," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2023–2026, 2015.

[9] P. Podder, S. Bharati, M. R. Mondal, P. Paul, and U. Köse, "Artificial neural network for cybersecurity: A comprehensive review," 06 2021.

[10] E. Grossi and M. Buscema, "Introduction to artificial neural networks," *European journal of gastroenterology & hepatology*, vol. 19, pp. 1046–54, 01 2008.

[11] K. Ghanem, F. J. Aparicio-Navarro, K. G. Kyriakopoulos, S. Lambotharan, and J. A. Chambers, "Support vector machine for network intrusion and cyber-attack detection," in *2017 Sensor Signal Processing for Defence Conference (SSPD)*, pp. 1–5, 2017.