

Comparative Analysis of Lightweight transformer models for malware classification on IoT devices

Dhanya L¹, Nandhana D²

¹Assistant Professor, Marian Engineering College, Trivandrum

²III Yr,BTech, Amrita School of Computing, Kollam

Abstract—As the Internet of Things (IoT) continues to expand, the threat posed by malware targeting IoT devices is becoming increasingly significant. This paper discusses three lightweight transformer models Squeezed ResNet (SResNet), TabNet, and MobileNet for classification in IoT data for malware detection. MobileNet has exceptional efficiency, making it perfect for low-power IoT applications; TabNet is excellent in interpretability and feature selection; and SE-ResNet strikes a balance between accuracy and resource consumption. This study highlights the significance of customizing designs to the particular limitations and needs of IoT systems and offers insightful information on how to choose the best models for IoT malware detection.

Index Terms— Internet of Things,SE- ResNet, TabNet, MobileNet.

I. INTRODUCTION

With billions of linked gadgets now able to interact and carry out activities on their own, the Internet of Things (IoT) has completely changed modern technology. Although this connectivity has spurred improvements in many sectors, it has also resulted in serious security flaws. Because they sometimes have low processing power and few security features, IoT devices are now popular targets for advanced malware attacks. For IoT systems to be dependable and secure, these dangers must be identified and mitigated.

Conventional malware detection techniques rely on traditional machine learning algorithms or rule-based techniques. Although somewhat successful, these techniques are unable to keep up with the constantly changing malware landscape and the complex, high-dimensional data produced by IoT networks. Conventional deep learning models are

computationally demanding and inappropriate for Internet of Things devices with limited resources. This makes light weight transformer-based models and other effective, and scalable solutions necessary. Transformer-based models, which are well-known for their capacity to handle a variety of data modalities and capture long-range relationships, have become a viable option for IoT malware detection in order to overcome these difficulties.

Transformer models are appropriate for IoT traffic analysis since they were first created for natural language processing jobs and have now been modified to work with structured and tabular data. These models improve malware detection capabilities by using self-attention mechanisms to find complex patterns and linkages in high-dimensional data. They also meet the requirements of IoT systems, which regularly generate big data streams, due to their scalability and ability to process large datasets. The application of transformer models in IoT malware detection is examined in this paper, with a focus on their high accuracy, interpretability, and computational economy. Their main strengths are the capacity to handle feature sparsity, real-time detection, and flexibility in limited settings. Examples of these models are TabNet, MobileNet and SE-Resnet. These cutting-edge architectures can be used to give IoT systems strong and proactive defenses against malware attacks.

M. N. Zaza[1] created a system that can identify malicious activity on the net-work from a particular IoT node. Using a central node that may be put within the network, malware on IoT can be detected through convolutional neural networks and monitoring. The accomplishment dispels the stigma associated with deep learning methods and demonstrates how these models can be readily

adapted and applied to any network. Khan [2] proposed, "Feature engineering-less-ML (FEL-ML)," which is lightweight and has less calculations on feature extraction. It efficiently speeds up the IoT edge with low power. Unprocessed packet byte-streams are used to train it. In addition to yielding superior outcomes, it is faster than conventional feature-based techniques. With the extra advantage of removing the need for substantial feature engineering investment by subject matter experts, FEL-ML enables resource-sensitive network traffic security. Sidharth [3] proposed a MALITE, a lightweight malware analysis system that can differentiate between harmful and benign binaries and classify different malware families. MALITE uses computationally cheap malware analysis techniques and low memory and battery power consumption to turn a binary into an RGB or grayscale image. MALITE-HRF is an ultra-lightweight random forest-based technique that makes use of sliding window-extracted histogram features, and MALITE-MN, a lightweight neural network-based architecture.

II. METHODOLOGY

The motivation of this work is due to the constraints in IoT devices highly optimized models are required due to limited compute power, memory, and energy availability. In order to detect IoT malware, it is frequently necessary to analyze sequential or high-dimensional data like as network traffic or logs. The attention mechanisms of transformers allow them to handle such data with ease. Lightweight transformers can be expanded to support a variety of Internet of Things devices, ranging from low-power sensors to high-end gateways.

A. Squeezed ResNet

ResNet or Residual Network is a deep learning architecture in which the layers learn residual functions with reference to the input layers[9][10]. Residual Blocks were a concept proposed by this design to solve the vanishing gradient problem. Skip connections are the method employed in this network. The skip connection skips some layers in between layer activations to link them to subsequent levels. This creates a block of residuals. By stacking these leftover bricks together, resnets are created. A lightweight version of the ResNet (Residual

Network) architecture designed to effectively handle tabular or structured data is called SE-ResNet (Squeezed and Excitation ResNet). It works especially well in situations with limited computational resources, like Internet of Things environments. Rather than having layers learn the underlying mapping, the method used in this network lets the network fit the remaining mapping.

An input convolutional block and its current channel count are provided to the function. Average pooling is used to reduce each channel to a single numerical value. The required nonlinearity is added using a ReLU function after a fully connected layer. There is also a certain proportional reduction in the complexity of its output channel. Each channel has a smooth gating function which is a Sigmoid activation after a second fully linked layer. The outcome of the side network is used to weight each feature map of the convolutional block.

It has Squeeze-and-Expand Modules to preserve feature richness while lowering the number of parameters, these modules substitute more effective "squeeze" (1x1 convolutions) and "expand" (1x1 and 3x3 convolutions) operations for conventional convolutional layers.

The residual Connections, similar to ResNet, are used to reduce the vanishing gradient problem and permit gradient flow, making deeper network training possible.

SE-ResNet lowers the total computing cost by employing parameter-efficient methods and squeeze operations.

The Squeeze phase each channel is compressed into a single numeric value so that the SE block has a global awareness of it. Global average pooling across each channel's spatial dimensions is done in this method, producing a vector of size "n," where "n" is the number of channels in the input tensor. In excitation phase, The vector of size "n" is then sent into a feed-forward neural network with two layers. This network is made to efficiently represent the complex interactions and dependencies between the channels. This network's output is once more a vector of size "n," with "n" values representing the learned significance weights for every channel. In scale and combine phase, each channel of the input tensor is scaled using the "n" weights that were acquired from the excitation phase. The SE block successfully suppresses less pertinent information and highlights

the most significant aspects by applying these adaptive weights to the original channels.

B. Tabnet

TabNet is essentially an algorithm that combines the advantages of DNN and tree learning techniques. Tree-based learning is the method used by TabNet for training. Through feature importance calculations, this approach not only improves the model's performance but also enables us to analyze the model's output. TabNet can learn representations on unlabeled input because it is a DNN-based model. It can efficiently locate the decision-boundary on the manifold by relying solely on the selected features, which is especially useful for tabular data with sparse features[11][12].

Tabnet employs a sequential attention mechanism in its deep tabular data learning architecture, to determine which attributes to consider at each decision stage. Feature masking, an attentive transformer, and a feature transformer make up the TabNet encoder. A split block separates the processed representation for the overall output and for use by the attentive transformer of the next phase. Global feature important attribution can be obtained by aggregating the feature selection masks, which offer interpretable information on the model's functionality at each step.

The main purpose of the TabNet model's encoder portion is to process the input features through a sequence of decision-making processes, focusing on the characteristics that are most pertinent to the prediction goal. TabNet's encoder updates the feature masking to be utilized in the subsequent encoder by using the result value of the previous encoder as feedback. Similar to the ensemble of trees in a tree-based model, this structure is essentially an ensemble of encoders. The feature transformer uses a variety of FC layers, BN, and Gated Linear Units (PV) to improve the feature representation inside the TabNet framework. Together, these layers map input features into an encoded space that is more helpful. In this situation, normalization—more especially, normalization by a factor of $\sqrt{0.5}$ is essential because it helps stabilize the learning process by preserving uniform variance throughout the network.

In order to guarantee sparsity in the feature selection, the attentive transformer then uses a learnable mask,

utilizing the *sparsemax* function. The mathematical representation of this mask is:

$$M[i] = \text{Sparsemax}(Z[i] - a[i-1]) \quad (1)$$

The feature transformer block forms the building block of the decoder section of the Tabnet. The feature transformer block uses a 4-layer network, of which two are dependent on the decision step and two are shared across all decision phases. Each layer is constructed with a fully connected layer, Batch Normalization and a GLU. An example of an attentive transformer block is when a single layer mapping is modulated using prior scale data that aggregates the amount of use of each feature before to the present decision step. The sparse max function selects the dominant features after the normalization of coefficients.

C. MobileNet

The MobileNet architecture is made to support mobile and embedded applications with great performance and efficiency. Effective convolutional neural network architectures for embedded and mobile vision applications make up the MobileNet family [4][5]. To lower the number of parameters and computational complexity and make it appropriate for devices with limited resources, the initial MobileNet architecture used depthwise separable convolutions.

The simplified architecture of MobileNet V2 includes:

A typical convolution layer with a stride of two and 32 filters. A Series of Inverted Residual Blocks in which each step of the network has a certain number of inverted leftover blocks. To control the computational complexity and receptive field, the strides, output channels, and expansion factors change from stage to stage. A global average pooling layer comes after the final convolution layer, which is a 1x1 convolution layer with 1280 filters.

A Fully Connected Layer is used for classification problems with softmax activation.

MobileNetV2 adds improvements including linear bottlenecks and inverted residuals. In order to filter features non-linearly, intermediate expansion layers use lightweight depthwise convolutions, while bottleneck levels are connected via an inverted residual structure. Both efficiency and performance are enhanced by this approach.

MobileNet and SE-ResNet are examples of convolutional architectures. They must be modified for tabular data in order to change the input layer from the image to one that corresponds to the dimensionality of your tabular data (number of features). An optional layer for embedding to make use of embeddings for any categorical features in the dataset. The Convolutional Layers is used for sequential or spatial tabular data, use 1D convolutions if you wish to preserve the convolutional structure. The convolutional outputs are flattened before putting them through tabular data-appropriate dense layers.

D. Transfer Learning of SE-ResNet and MobileNet

MobileNet and SE-ResNet are examples of convolutional architectures. They must be modified for tabular data. The input layer should be changed from the image to one that corresponds to the dimensionality of your tabular data (number of features). An optional layer for embedding Make use of embeddings for any categorical features in the dataset. Convolutional Layers are used for sequential or spatial tabular data. 1D convolutions are used to preserve the convolutional structure. The convolutional outputs are flattened before putting them through tabular data-appropriate dense layers. The pretrained weights from MobileNet or SE-ResNet are used if working with structured data that has spatial dependencies, these may not be immediately helpful for tabular data. As an alternative, concentrate on modifying the architecture and training it either from the ground up or with a limited number of pretrained weights.

During the training phase of transfer learning only the last layers of the network should be updated; other layers, such as the early layers, should be frozen.

The tabular dataset is fine-tuned with the entire model after gradually unfreezing the earlier layers. Cross entropy is used as the loss function for classification and the Adam optimizer is employed. Learning rate scheduling increases the effectiveness of training.

which includes malware and benign data. It has network metrics and biometric parameters. The software used is Python 3.1 on Jupyter Notebook. The problem is that the dataset has labels as 0 and 1 where 0 represents the benign data and 1 representing the malware. To evaluate the performance of each model metrics like accuracy, precision, recall and F1-score are computed.

True Positive (TP): The actual result was positive, as the model had accurately predicted.

True Negative (TN): The actual result was negative, as the model had accurately anticipated.

False Positive (FP): When the model expected a positive result, it produced a negative result instead.

False Negative (FN): When a positive result was actually obtained, the model mispredicted a negative result.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (4)$$

$$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

Table 1. Performance metrics of various models on the Data Set

Metrics	Benign/malware	Light weight Models		
		SE-Resnet	MobileNet	Tabnet
Precision	0	0.94	0.95	0.91
	1	0.94	0.95	1.00
Recall	0	0.99	0.98	1.00
	1	0.56	0.67	0.41
F1-Score	0	0.97	0.98	0.96
	1	0.70	0.63	0.57
Accuracy		0.94	0.99	0.92
Cohen's Kappa		0.890	0.997	0.5339

III. RESULTS AND DISCUSSIONS

The dataset used is WUSTL EHMS 2020[13] Dataset for Internet of Medical Things (IoMT) Cybersecurity

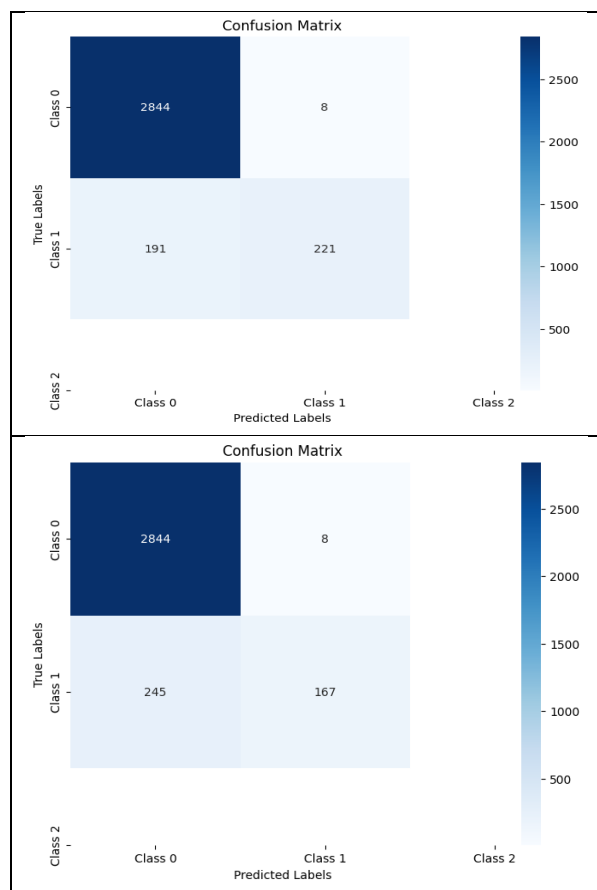


Fig 1. Confusion matrix of SRResNet and TabNet in malware classification

A matrix that depicts how well a machine learning model performs on a set of test data is called a confusion matrix. Based on the model's predictions, it shows the proportion of accurate and wrong occurrences.

A. Resource related metrics

The memory capacity (measured in MB/GB) needed to store the model's settings. It is Essential for deployment on systems with little storage capacity or edge devices. MobileNet is compact, designed with mobile and embedded devices in mind. SE-ResNet has more SE modules make it bigger. TabNet is moderate due to its attention-based feature learning, it is moderately big.

.Latency is the amount of time required to predict a single input or batch. It has real-time usability, particularly in low-latency applications such as autonomous systems or the Internet of Things. Because MobileNet uses depthwise separable convolutions, it is the fastest. Due to its increased

parameter count and SE blocks, SE-ResNet is slower. The sequential judgment processes of TabNet can lengthen the inference time; with a moderate latency.

Training time is the total amount of time spent on a set number of epochs of model training. It shows viability in situations requiring frequent retraining or time constraints. MobileNet has less parameters with faster training. SE-ResNet is Slower because of its SE blocks and deeper design. TabNet takes a fair amount of time to train, particularly when using bigger datasets.

Scalability is the capacity to sustain effectiveness or performance as input or data dimensions increase. MobileNet has a light weight architecture and hence scalable. SE-ResNet is constrained for bigger datasets due to deeper layers. TabNet can be adjusted for high dimensional tabular data.

Interpretability is the explanation of the model judgements. MobileNet is highly efficient but has less interpretability. Though SE-resnet has feature enrichment, the interpretability is lesser. Due to the feature relevance and the attention process TabNet is highly interpretable.

Table 2. Parametric metrics of various models

Metric	MobileNet	SE-ResNet	TabNet
Model Size	3 MB	100 MB	2 MB
Inference Time	10 ms	15 ms	5 ms
FLOPs	300 MFLOPS	4 GFLOPS	<100 MFLOPS

The performance of light weight transformer models with the conventional deep learning models are listed in Table 2.

The light weight transformer models provide better accuracy when compared to Convolutional neural networks and Long Short term Memory or Gated recurrent Unit which are predominately used for classification tasks. Though the conventional transformer models are significantly resource intensive, their light weight versions are highly optimised and hence can be efficient for resource constrained devices. Moreover the transformer models can handle larger datasets and the complex

interaction between the features when compared to CNN and LSTM. The light weight transformer models are highly robust when compared to the other deep learning models.

IV. CONCLUSION

We compare transformer-based methods for IoT malware detection in this paper, emphasizing their effectiveness, resource efficiency, and suitability for actual IoT situations. In order to improve IoT security and tackle the increasing difficulties brought on by cyberthreats in this field, this study intends to demonstrate the revolutionary potential of these models. MobileNet is the best option for real-time applications because of its effectiveness and quickness. TabNet is unique for applications that prioritize interpretability since it provides both transparency and accuracy. Because of its strong feature extraction capabilities and exceptional accuracy, SE-ResNet is advised for centralized IoT systems with substantial processing resources.

REFERENCES

- [1] A. M. N. Zaza, S. K. Kharroub and K. Abualsaud.: "Lightweight IoT Malware Detection Solution Using CNN Classification," 2020 IEEE 3rd 5G World Forum (5GWF), Bangalore, India.doi: 10.1109/5GWF49715.2020.9221100.(2020).
- [2] Khan, Arshiya and Chase Cotton.: "Efficient Attack Detection in IoT Devices using Feature Engineering-Less Machine Learning.:" ArXiv abs/2301.03532 (2022).
- [3] Anand, S., Mitra, B., Dey, S., Rao, A., Dhar, R., & Vaidya, J. MALITE: Lightweight Malware Detection and Classification for Constrained Devices(2023).
- [4] Rybczak, M., & Kozakiewicz, K. : Deep Machine Learning of MobileNet, Efficient, and Inception Models. Algorithms, 17(3), 96.https://doi.org/10.3390/a17030096.(2024)
- [5] Sang, X., Ruan, T., Li, C. et al. :A real-time and high-performance MobileNet accelerator based on adaptive dataflow scheduling for image classification. J Real-Time Image Proc 21, 4 . https://doi.org/10.1007/s11554-023-01378-5.(2024)
- [6] Odabas, Ceyla & Kahramanoğlu, Recep.: Analyzing the Effects of Mobile Devices on Student Achievement: A Wide- ranging Meta-analysis. International Online Journal of Educational Sciences. 5. 1018-1034.(2023)
- [7] Liang, Jiazhi. : Image classification based on RESNET. Journal of Physics: Conference Series. 1634.(2020)
- [8] Justice Kwame Appati, Isaac Adu Yirenkyi.:A cascading approach using se-resnext, resnet and feature pyramid network for kidney tumor segmentation,Heliyon,Volume 10, Issue 19.e38612, ISSN 2405-8440, https://doi.org/10.1016/j.heliyon.2024.e38612.(2024)
- [9] Shanu Nizarudeen, Ganesh Ramaswamy Shanmughavel.: Comparative analysis of ResNet, ResNet-SE, and attention-based RaNet for hemorrhage classification in C images using deep learning,Biomedical Signal Processing and Control, Volume 88, Part A,105672,ISSN 1746-8094, https://doi.org/10.1016/j.bspc.2023.105672.(2024)
- [10]He Gong, Lin Chen, Haohong Pan, Shijun Li, Yin Guo, Lili Fu, Tianli Hu, Ye Mu, Thobela Louis Tyasi,Sika Deer.: Facial Recognition Model Based on SE-ResNet,Computers, Materials and Continua,Volume 72, Issue 3,Pages 6015-6027, ISSN 1546-2218. https://doi.org/10.32604/cmc.2022.027160.(2022)
- [11]Yingze, S., Yingxu, S., Xin, Z. et al.: Comparative analysis of the TabNet algorithm and traditional machine learning algorithms for landslide susceptibility assessment in the Wanzhou Region of China. Nat Hazards 120, 7627–7652 (2024).
- [12]Ta, V. C., Hoang, T. L., Doan, N. S., Nguyen, V. T., Nguyen Dieu, N., Pham, T. T. T., & Nguyen Dang, N.:Tabnet efficiency for facies classification and learning feature embedding from well log data. Petroleum Science and Technology, 1–16, https://doi.org/10.1080/10916466.2023.2223623. (2023)
- [13]WUSTL EHMS 2020 Dataset for Internet of Medical Things (IoMT) Cybersecurity Research.

- [14] Zhen, Y., & Zhu, X. :An Ensemble Learning Approach Based on TabNet and Machine Learning Models for Cheating Detection in Educational Tests. Educational and Psychological Measurement (2024).