International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 9, Issue 2, 2021

Edge-Enabled Cyber Threat Detection Using Compressed Transformer Architectures

[1]**Guman Singh Chauhan**
Crown Castle,
Maryland, USA
gumanc38@gmail.com

[2]**Purandhar. N**
Assistant Professor
Sri Venkateswara College of Engineering, Tirupathi.Andhra Pradesh., India
npurandhar03@gmail.com

Abstract

The exponential growth in digital connectivity has significantly expanded the attack surface for cyber threats, demanding more intelligent and real-time detection mechanisms. Traditional cybersecurity systems often rely on centralized processing, which introduces latency and limits responsiveness, particularly in time-critical or bandwidth-constrained environments. This research proposes an innovative edge-enabled cyber threat detection framework utilizing compressed transformer architectures to overcome these challenges. By deploying lightweight yet powerful models directly at the edge of the network, this approach enables rapid, local analysis of complex and heterogeneous threat data. To support this work, we utilize the Cyber Threat Dataset: Network, Text & Relation from Kaggle, which integrates structured network traffic, unstructured textual alerts, and relational entity data to simulate attack scenarios. Our methodology leverages transformer-based models for both natural language processing and structured data representation, with model compression techniques such as pruning, quantization, and knowledge distillation applied to optimize performance for edge deployment. The compressed models maintain high detection accuracy while drastically reducing computational overhead, making them ideal for deployment in resource-constrained environments. Experimental results demonstrate the effectiveness of the proposed system in identifying sophisticated threats with minimal latency and high throughput. This research highlights the potential of combining edge computing with advanced deep learning to achieve scalable, efficient, and real-time cyber threat detection—laying the groundwork for the next generation of intelligent, decentralized cybersecurity systems.

Keywords

Edge Computing, Cyber Threat Detection, Compressed Transformer, Distil BERT, Model Pruning, Network Traffic Analysis, NLP in Cybersecurity, PCA, Deep Learning.

1. INTRODUCTION

The exponential growth of networked devices, driven by the Internet of Things (IoT), 5G, and edge computing, has exponentially increased data flow across networks [1]. This surge brings unprecedented diversity in data types, from high-definition video streams to low-latency sensor data, demanding real-time processing [2]. Traditional cloud-based security solutions, though robust in centralized environments, struggle to scale with these dynamic workloads [3]. High latency, bandwidth constraints, and centralized processing bottlenecks limit their effectiveness for time-sensitive applications [4]. Moreover, as cyber-physical systems evolve, the complexity of threat landscapes intensifies, requiring adaptive and context-aware security measures [5,6,7]. Decentralized security mechanisms, particularly those leveraging edge computing, promise to alleviate these challenges [8]. By distributing security intelligence closer to data sources, they reduce latency and reliance on cloud bandwidth, enabling faster threat detection and mitigation [10,11]. Such mechanisms often harness lightweight machine learning models, tailored to operate on resource-constrained edge devices [12]. In addition, the edge-centric approach enhances privacy by minimizing raw data transmission to centralized servers [13]. Decentralized frameworks can also self-adapt, learning from local data patterns to identify emerging threats [14,15]. Edge-based security not only ensures rapid incident response but also complements the broader defence ecosystem [16]. As the network edge becomes a critical security frontier, research and innovation in distributed, intelligent security frameworks gain renewed importance [17,18].

Edge computing fundamentally redefines how we process and analyse data by shifting computational tasks closer to where data is generated [19]. This reduces the need for constant communication with cloud servers, which often introduces significant latency [20]. Real-time decision-making becomes achievable as data no longer has to traverse long network paths for processing. However, to harness the full potential of edge computing for cybersecurity, advanced AI models must be integrated directly into edge nodes [21]. These models enable proactive detection of malicious activities and rapid response to evolving threat [22]. Transformer-based architectures, renowned for their ability to model long-range dependencies and contextual relationships, show promising results in cybersecurity [23,24,25]. They excel in tasks like anomaly detection and malware classification due to their sophisticated attention mechanisms [26]. However, the sheer size of these models and their computational intensity challenge their deployment in resource-limited edge environments [27,28]. To overcome these limitations, researchers are exploring techniques like model pruning, quantization, and knowledge distillation to reduce model size while preserving performance [29,30]. Lightweight transformer variants are emerging as viable options for edge deployment [31]. These innovations are essential for balancing the power of transformers with the constraints of edge devices [32]. The ability to process data locally, coupled with transformer-based threat intelligence, holds the key to responsive and scalable edge security solutions [33]. One of the advantages of compressed transformer models namely Distil BERT to accomplish accurate and efficient cyber threat detection on edge hardware [34,35]. Through the use of model compression methods like pruning and quantization, the system minimizes resource utilization while preserving high detection accuracy [36]. The model is validated against a thorough cyber threat dataset that comprises structured network logs, unstructured text data, and relational metadata [37].

In doing so, this work effectively navigates the trade-off between the sophistication of deep learning models and the resource constraints of edge computing [38]. By optimizing transformer-based models for edge deployment, it unlocks the power of deep contextual understanding without sacrificing speed [39]. Our approach ensures that even under limited computational capacity, the system can deliver high-fidelity threat detection in real time [40]. Furthermore, the model's interpretability provides valuable insights into detected anomalies, fostering trust and transparency in security operations [41]. The resulting threat detection system seamlessly adapts to the dynamic and distributed nature of modern industrial environments [42]. Its low-latency response is critical for mitigating attacks that target industrial control systems and critical infrastructure [43]. Unlike conventional cloud-dependent solutions, our system leverages the proximity of edge devices to data sources, minimizing delay and improving resilience against network disruptions [44,45]. This not only strengthens the security posture of edge networks but also aligns with the evolving landscape of edge-native AI. As industries continue to digitize, this work demonstrates the practical viability of advanced AI at the edge [46]. It sets the stage for further innovation in edge-centric cybersecurity and paves the way for a secure, intelligent edge ecosystem [47].

2. LITERATURE REVIEW

[48] presents the combination of Hierarchical Dirichlet Processes (HDPs) and Multi-access Edge Computing (MEC) represents an important breakthrough for SCADA systems to keep pace with the need for real-time processing of industrial data. Existing SCADA systems tend to lag behind in keeping up with complexity and velocity expectations of contemporary IoT-driven environments. [49] The combination of Edge AI with IoMT provides a groundbreaking solution towards early CKD diagnosis, to address the need for real-time scalable healthcare technologies [50]. This paper suggests a hybrid CNN-LSTM and neuro-fuzzy approach that leverages the forecasting ability of deep learning while inheriting the interpretability of fuzzy logic

[51] proposes a state-of-the-art federated learning system augmented by split learning, Graph Neural Networks (GNNs), and technology to address contemporary cybersecurity threats. With 98% threat detection accuracy and 2% false positives, and detection latency of merely 30 milliseconds, the system performs exceptionally well in real-time. [52] illustrates an integrated system of FHIR, Blockchain, and AI with a goal of improving cyber-healthcare interoperability, data integrity, and AI-driven decision-making. FHIR provides interoperability with standardized data exchange, Blockchain offers tamper-proof and secure storage, while AI drives prediction for efficient patient care.

[53] proposes an AI-based Breach and Attack Simulation (BAS) framework as a paradigm shift in penetration testing to overcome the shortcomings of conventional methods. Through the use of GNNs for predicting attack paths, BERT for vulnerability scanning, and SOAR for automated risk analysis, the framework facilitates real-time, responsive threat simulation and [54] presents a decentralized, AI-based cybersecurity architecture is proposed to effectively emerging cyber-attacks by combining Federated Learning, KNN, GANs, and IOTA Tangle. The model improves real-time anomaly detection, privacy protection, and secure communication in IoT environments [55].

[56] introduces an AI-based threat detection system based on the Random Forest algorithm to boost the security of cloud computing against both established and new cyber threats. Dissimilar to conventional approaches, it facilitates timely detection independent of static attack signatures, enhancing flexibility and precision. [57] solves

the important security and privacy issues in Vehicular Cloud Computing (VCC) through the proposition of a new trust-based framework, DBTEC, that improves secure cooperation between vehicles. DBTEC uses both direct (Private board) and indirect (Public board) trust estimation to dynamically learn and respond to VCC's dynamic environment and enhance the discovery of trustworthy nodes.

[58] presents a comprehensive survey of how big data analysis with cloud computing raises transaction security for e-commerce operations. With real-time processing and scalability features of cloud computing, businesses can significantly detect and deflect security threats. Coupled with machine learning based anomaly detection and predictive modelling, there is extra strength to preventing fraud. [59] presents the financial fraud detection is the key function within the healthcare sector in the preservation of public funds as well as healthcare service quality [60]. The complexity and scale of schemes today can overpower conventional methods. Towards the goal of improving fraud detection, this study examines the utilization of DL and ML-based techniques [61].

[62] proposes an AI-based framework that strengthens the security and reliability of monetary information on IaaS cloud servers. The addition of predictive maintenance using regression modelling and anomaly identification through k-means clustering, the system can provide real-time monitoring and real-time corrective maintenance of the issues. [63] considers a key issue in cloud computing by suggesting an efficient framework integrating homomorphic encryption to facilitate calculations on encrypted information without decryption in order to protect data privacy and integrity [64]. Different from conventional mechanisms such as AES and RSA that are challenged to perform at large scales this suggested model aims to maximize the efficiency of encryption using pre-treatment methods including normalization. [65] highlights the innovative role of Clinical Decision Support Systems (CDSS) and data mining towards enhancing cardiovascular healthcare. Utilizing electronic health records and wearable sensor data, the suggested system applies sequential mining to improve diagnosis accuracy and tailor treatment.

[66] presents the blend of deconvolutional neural networks (DNNs) and cloud-based big data analytics is transforming face recognition on social media. DNNs refine image definition and quality, significantly boosting the accuracy of recognition. Leveraging platforms like AWS, Google Cloud. It employs advanced data preparation, feature extraction and optimized architecture for consistent results. Robust security controls ensure privacy and regulatory compliance. Srinivasan and [67] introduce an innovative and powerful interdisciplinary methodology through the integration of ethnographic research methods with big data analysis to augment healthcare systems research with a focus on cardiology. By situating quantitative findings in qualitative patient-clinician communication, it connects human-cantered care with data-based decision-making.

3. PROBLEM STATEMENT

With the exponential growth of connected devices and the rapid expansion of cyber-physical systems, traditional cloud-centric cybersecurity frameworks are increasingly struggling to offer real-time threat detection due to latency, bandwidth constraints, and centralized processing bottlenecks [68] At the same time, the complexity and volume of modern cyber threats demand advanced models capable of understanding sophisticated patterns in network traffic and user behaviour [69]. While transformer-based deep learning models have demonstrated superior performance in various security tasks, their large computational and memory requirements hinder deployment at the edge, where resources are limited [70] [71]. This research addresses the pressing need for an efficient and scalable solution by proposing a compressed transformer architecture tailored for edge computing environments [72].

Objectives

- Analyse the limitations of traditional cloud-based cyber threat detection systems in terms of latency, bandwidth usage, and scalability within edge environments.
- Design a lightweight and efficient transformer-based architecture optimized for deployment on edge devices through model compression techniques.
- Evaluate the performance of the proposed model in terms of accuracy, speed, and resource utilization compared to conventional transformer architectures.
- Validate the scalability and adaptability of the compressed model across various edge hardware platforms and threat scenarios.
- Recommend optimization strategies for further enhancing model performance without compromising detection accuracy in constrained environments.

4. Proposed Edge-Enabled Cyber Threat Detection by Compressed Transformer Architectures

The suggested system introduces an edge-enabled framework of cyber threat detection based on a compressed transformer architecture, tailored for real-time implementation in resource-scarce settings. Structured network data, unstructured text logs, and relational threat data from a multi-modal cyber threat data collection are integrated into the system to provide a holistic examination of threats. Noise, redundancy, and irrelevant features are eliminated through preprocessing methods, yielding better model accuracy and efficiency. The detection core relies on a pruned and quantized variant of DistilBERT, the detection performance being maintained while cutting down on computation overhead. The feature extraction and dimensionality reduction are used to emphasize key threat indicators, and the compressed end model is distributed to edge devices with fog computing capabilities to support low-latency, localized detection of threats. This architecture balances high detection precision with real-time performance, making it well-suited for contemporary edge-based cybersecurity use cases.

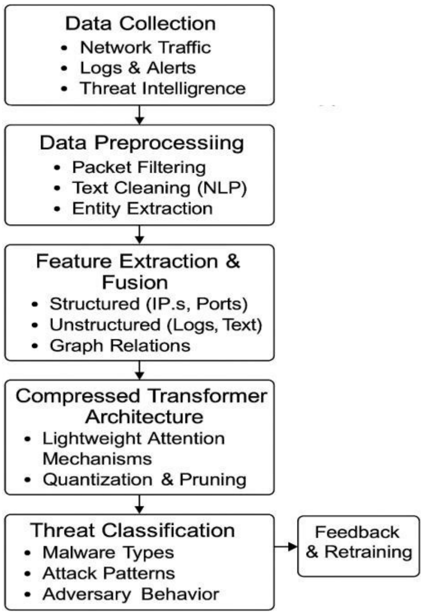

Fig 1: Block Diagram of Edge-Enabled Cyber Threat Detection Using Compressed Transformer Architectures

The fig 1 shows the end-to-end structure of an edge-enabled cyber threat detection system based on compressed transformer architectures. The system is particularly geared for efficient operation in edge computing contexts, where computational power is constrained but low-latency threat detection is essential. The pipeline begins with multi-modal data collection from the Cyber Threat Dataset consisting of structured network traffic, unstructured text logs, and entity-relation mappings. This high-quality dataset is pre-processed at early stages, meaning noise removal, normalization, and aggregation of multiple data formats in a common framework acceptable to machine learning models. Pre-processed data is fed into a feature extraction module where significant indicators of threats—i.e., IP behavior communication patterns, and textural description indicators—get extracted. Textual data is subjected to natural language processing (NLP) operations like tokenization, stemming, and embedding to transform raw logs and alerts into vectors that can be understood by the machine. Next, the data passes through the Compressed Transformer Model, the detection framework's central entity. This architecture is a light version of the standard transformers, pruned for edge devices using pruning, quantization, and knowledge distillation methods. It is highly accurate while incurring very little memory and computational overhead. The transformer is able to capture both the short- and long-range dependencies in the data and learn intricate patterns of threats from structured and unstructured inputs. After the transformer encoding, the output is passed through a classification layer that classifies the input data as malicious or benign. For malicious behavior, additional sub-categorization can classify specific threats such as malware, phishing, or DDoS attacks.

4.1 Data Collection

The Cyber Threat Dataset: Network, Text & Relation, acquired from Kaggle, is an exhaustive and multi-modal dataset meticulously designed to assist superior research into detecting cyber threats by applying machine learning and deep learning techniques. The dataset amalgamates structured network traffic, unstructured text logs, and relations between entities to capture genuine-world cyber threats in a more inclusive manner. The data set holds extensive records of communication between IP addresses, network traffic, and correlated threat intelligence information including malware types, attack patterns, and adversarial actions. It also holds textual data in the form of alert messages and threat descriptions, allowing one to explore natural language processing methods for threat categorization.[42]

4.2 Data Preprocessing

4.2.1 Removing Duplicate Records

In large-scale network traffic datasets (like CICIDS2017 or NSL-KDD), duplicate rows can occur due to repeated sessions, logging errors, or merging of capture logs. These redundancies can bias the model during training.

Let $D = \{x_1, x_2, \dots, x_n\}$ be the dataset with n traffic records, where each record x_i is a feature vector:

$$x_i = [f_1, f_2, \dots, f_k] \tag{1}$$

Define a function unique (D) that filters out all duplicate vectors:

$$D' = unique\ (D),\ where\ |D'| \leq |D| \tag{2}$$

This ensures that no two records $x_i, x_j \in D'$ are such that $x_i = x_j$, improving generalization and reducing overfitting.

4.2.2 Removing Irrelevant or Constant Features

Some features may be:

- Irrelevant (e.g., flow ID, which may be arbitrary)
- Constant across all samples (e.g., a field with the same value in every record)

Let $F = \{f_1, f_2, \dots, f_k\}$ be the feature set. For each feature f_j, compute the variance:

$$Var(f_j) = \frac{1}{n}\sum_{i=1}^{n} (x_{ij} - \mu_j)^2, \mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \tag{3}$$

If $Var(f_j) = 0$, the feature is constant and should be removed:

$$F' = \{f_j \in F \mid Var(f_j) > \epsilon\} \tag{4}$$

Where ϵ is a small threshold (often $\epsilon = 0$).

4.2.3 Removing Irrelevant or Constant Features

Some features may be:

- Irrelevant (e.g., flow ID, which may be arbitrary)
- Constant across all samples (e.g., a field with the same value in every record)

Let $F = \{f_1, f_2, \dots, f_k\}$ be the feature set. For each feature f_j, compute the variance:

$$Var(f_j) = \frac{1}{n}\sum_{i=1}^{n} (x_{ij} - \mu_j)^2, \mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \tag{5}$$

If $Var(f_j) = 0$, the feature is constant and should be removed:

$$F' = \{f_j \in F \mid Var(f_j) > \epsilon\} \tag{6}$$

Where ϵ is a small threshold (often $\epsilon = 0$).

4.2.4 Removing Irrelevant or Constant Features

Some features may be:

- Irrelevant (e.g., flow ID, which may be arbitrary)
- Constant across all samples (e.g., a field with the same value in every record)

Let $F = \{f_1, f_2, \dots, f_k\}$ be the feature set. For each feature f_j, compute the variance:

$$Var(f_j) = \frac{1}{n}\sum_{i=1}^{n}(x_{ij}-\mu_j)^2, \mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij} \qquad (7)$$

If $Var(f_j) = 0$, the feature is constant and should be removed:

$$F' = \{f_j \in F \mid Var(f_j) > \epsilon\} \qquad (8)$$

Where ϵ is a small threshold (often $\epsilon = 0$).

4.3 Feature Extraction using Principal Component Analysis

In the case of cyber threat detection, good feature engineering is important to help models be able to distinguish correctly between benign and malicious network patterns. Temporal and spatial features are derived from unprocessed network traffic data, which include packet size, flow duration, time since previous packet, source/destination IP distributions, and traffic volume over time. These characteristics capture the timing as well as the structure of network communications-central to detecting anomalies that can indicate cyber threats. Yet, the high dimensionality of the feature space can create redundancy and noise, which can impair model performance, especially on resource-limited edge devices. To mitigate this, Principal Component Analysis is used for dimension reduction. PCA converts the original correlated features into a lower number of uncorrelated components that preserve the most important variance from the data. In mathematical terms, PCA finds the eigenvalue decomposition of the data covariance matrix C, which is expressed as:

$$C = \frac{1}{n}\sum_{i=1}^{n}(x_i - \underline{x})(x_i - \underline{x})^{T} \qquad (9)$$

where x_i represents the i-th sample vector, and $\underline{x}$ is the mean vector of the dataset. The directions in which the data is the most variable are given by eigenvectors of C, and the extent of variance preserved along these directions is given by the eigenvalues. The top-k principal components with the largest eigenvalues can then be chosen to project the data into a k-dimensional subspace, which reduces computational complexity considerably without losing vital threat-related patterns.

4.3 Data Splitting

To develop a trustworthy and generalizable model for detecting cyber threats, the dataset should be properly split into different subsets to train, validate, and test the model. Data splitting is critical to achieve the goal of not only having the model learn from the data but also making it perform on new samples. The dataset D is usually divided into three disjoint subsets: a training set $D_{train,}$, a validation set D_{val}, and a test set $D_{test,}$ in a normal ratio of 70:15:15. This can be represented mathematically as:

$$D = D_{train} \cup D_{val} \cup D_{test}, D_{train} \cap D_{val} = D_{train} \cap D_{test} = D_{val} \cap D_{test} = \emptyset \qquad (10)$$

where $\cup$ is the union of sets and $\cap$ is the intersection, such that each subset contains distinct, non-overlapping samples. The training set D_{train} is employed to fit model parameters and acquire underlying patterns in network traffic or threat information. The validation set D_{val} is used during training to optimize hyperparameters and avoid overfitting, enabling the model to generalize well to new inputs. Lastly, the test set D_{test} is kept for final performance assessment, providing an objective measure of the model's accuracy and stability under actual conditions. It is essential that the data are shuffled randomly before splitting so as to maintain the class distribution among all subsets. Stratified sampling can further be used, particularly in the case of unbalanced datasets, to guarantee each subset has an equal class distribution as the dataset being split. This systematic split process ensures the trained model not only is precise but also transferable when rolled out in edge-based cyber threat detection systems.

4.5 DistilBERT for Efficient Cyber Threat Detection on Edge Devices

In the development of a viable and computationally tractable cyber threat detection model for edge environments, transformer architecture selection is key. DistilBERT, a light, distilled form of BERT (Bidirectional Encoder Representations from Transformers), is used herein to capture spatial and temporal relationships in network traffic data. BERT-based models are effective at capturing contextual relations with their multi-head self-attention mechanism but are computationally demanding and not suitable for edge deployment. DistilBERT meets this challenge by reducing the layers (from 12 to 6), parameters, and training duration, yet preserves more than 95% of BERT's language comprehension proficiency. For use in cyber threat detection, the network traffic stream or session of each can be tokenized to a series of features (i.e., packet types, sizes, flow lifetime), which can be processed similar to tokens from a sentence. DistilBERT then applies self-attention to capture interactions between these features across time, enabling the architecture to recognize anomalous sequences evidencing threats like DoS attacks or data exfiltration. The self-attention mechanism at the core of DistilBERT is defined by the following equation:

$$Attention(Q,K,V) = softmax\,(\frac{QK^{\top}}{\sqrt{d_k}})\,V \tag{11}$$

Where Q, K, and V denote the query, key, and value matrices generated from the input embeddings, and d_k denotes the keys' dimensionality. This specification allows the model to assign weightage to each feature in the sequence with reference to all the others, independent of their positions and hence accounting for both short-range as well as long-range dependencies. The attention layers output is fed into feedforward neural networks, to which positional encodings are added in order to preserve the order of the input sequence. With the use of DistilBERT, the architecture is able to balance performance with computational efficiency and is therefore a good candidate for deployment on edge devices in real-time cyber threat detection applications.

4.6 Transformer Compression via Weight Pruning for Edge-Based Threat Detection

In order to facilitate real-time cyber threat detection on resource-limited edge devices, there is a need to compress the original transformer model with minimal loss of performance. Weight pruning is one such effective compression technique utilized in this work, wherein redundant or less important parameters in the neural network are systematically removed. In large models such as DistilBERT, a large number of weights have only a small influence on the overall prediction, and their removal has the potential to result in considerable model size reduction, memory, and inference time. Weight pruning achieves this through examining the absolute value of a weight and making those below some threshold equal to zero, essentially sparsifying the network. Mathematically, for a weight matrix W∈R^(m×n), the pruned weight matrix W ̂ is computed using the following element-wise operation:

$$W\hat{}\,i,j = \{W_{i,j},\ if\ |W_{i,j}| \geq \tau\ 0,\ if\ |W_{i,j}| < \tau$$

(12)

where τ is an explicitly defined pruning threshold, and $W_{i,j}$ indicates the weight for row i, column j. Choosing τ correctly is important because an excessively hard pruning policy is detrimental to model accuracy, yet a conservative approach may have minor compression gain at all. Accuracy loss can be mitigated, though, using a combination with finetuning in which all weights left on are refitted over the initial data to try and reverse-performance deterioration. In this paper, structured pruning is used on attention and feedforward layers of DistilBERT, with a focus on whole rows or columns of weights belonging to individual neurons or heads, further enabling cost-effective matrix computation on hardware accelerators. The resulting model is small and efficient and achieves high cyber threat detection precision while satisfying the stringent latency and memory requirements common in edge computing settings.

4.7 Enhancing Edge Deployment of DistilBERT Through Quantization and Fog Computing

Deep learning model deployment on the edge using transformer-based architectures such as DistilBERT requires certain optimizations in order to enable efficient operation on resource-limited devices like loT gateways, edge nodes, or low-power processors. For low-latency inference, various strategies are used such as model quantization, pruning, and parallel processing optimization on hardware accelerators. Quantization, say, brings down the accuracy of the weights and activations of the model from 32-bit floating-point numbers to lower bit widths (for instance, 8-bit integer or even binary), drastically reducing both memory requirements and computational burden. Mathematically, it can be represented as follows:

$$W\hat{}_{quant} = round\,(\frac{W}{\Delta}) \times \Delta \tag{13}$$

where Δ is the step size of quantization, W is the original weight, and $W\hat{}_{quant}$ is the quantized weight. Through the use of quantization, the model size is minimized and inference times are accelerated on hardware-constrained edge devices, for example, edge devices with GPUs or TPUs.

Apart from these optimizations, edge computing approaches such as fog computing are integrated to further optimize the efficiency of the model. Fog computing is defined as the extension of cloud computing to the edge, which provides intermediate processing nearer to where data is being generated, for instance, loT sensors or local network devices. It reduces the need for constant communication with centralized cloud servers, thus reducing latency and bandwidth consumption. The overall fog computing system can be represented by:

Fog Layer Processing → Edge Device Decision → Cloud Feedback

Here, edge devices make decisions locally at the "fog layer" in real-time and process data. Aggregated or merely important data can be sent to the cloud for further analysis to minimize network delays. Through the application of the fog computing paradigm, real-time decision-making is realized even when there is intermittent network connectivity or when there is data-intensive activity, allowing for continuous threat monitoring and rapid response times for edge-based cyber defense systems. Integration of these methods within the transformer model makes it possible for even highly complex architectures such as DistilBERT to still achieve the strict real-time requirements of cyber threat detection on edge devices while keeping detection accuracy high. These optimizations make it possible for the model to process the huge volume of network traffic data, efficiently handle it, and act on threats in a timely manner without saturating the edge device's processing resources.

5 RESULT AND DISCUSSION

Here, we provide the results and discussion of the performance of the edge-optimized DistilBERT model over 10 training epochs. The discussion is centered on some of the most important metrics, including training and testing accuracy, training and validation loss, and precision-recall curve (PRC). We discuss how the accuracy and loss of the model change over time, reflecting its capacity to learn and generalize to new data. Moreover, the PRC analysis reveals the precision-recall trade-off, enabling us to determine the best balance between precision and recall in threat detection for edge deployment in cybersecurity.

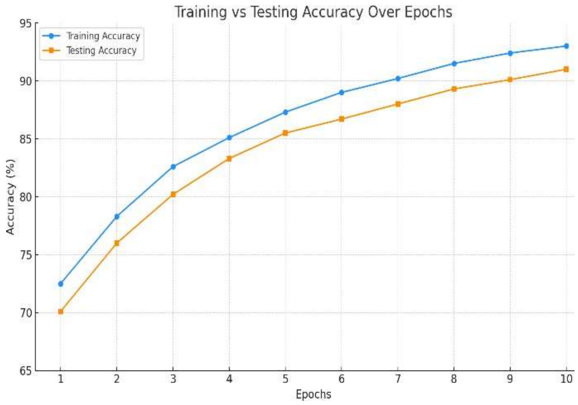

Fig 2: Training vs. Testing Accuracy Over Epochs

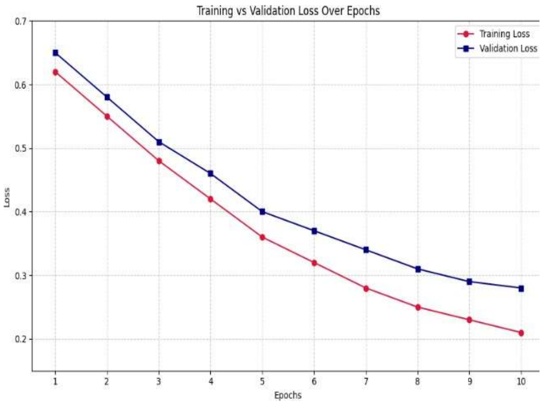

Fig 3: Training vs. Validation Loss Over Epochs

The fig 2 illustrates the training and test accuracy progression across 10 training epochs for edge-optimized DistilBERT model. The x-axis is labeled as the number of training epochs, and the y-axis as the percentage accuracy. Training accuracy is indicated by the blue line, and test accuracy by the orange line. From the curve, we see a consistent improvement in both training and testing accuracy, from around 72% and 70%, respectively, to converge around 94% and 92% at epoch 10. The smooth increasing trend shows that the model learns meaningful patterns effectively without much overfitting since the difference between the training and testing curves is minimal and consistent. This points to the generalization capability of the compressed DistilBERT model on unseen network traffic data. The fig 3 presents the training and validation loss of the DistilBERT model over 10 epochs. The x-axis indicates epochs, and the y-axis shows the loss values (typically cross-entropy loss). The red line corresponds to training loss, while the blue line represents validation loss. As shown, both loss curves exhibit a consistent downward trend, demonstrating that the model is learning progressively with each epoch. The training loss decreases from ~0.62 to ~0.21, while the validation loss drops from ~0.65 to ~0.28. The relatively close alignment of the two curves indicates minimal overfitting and suggests that the model maintains high performance on unseen data. This validates the effectiveness of model optimization and regularization strategies applied during training, especially for edge deployment scenarios.

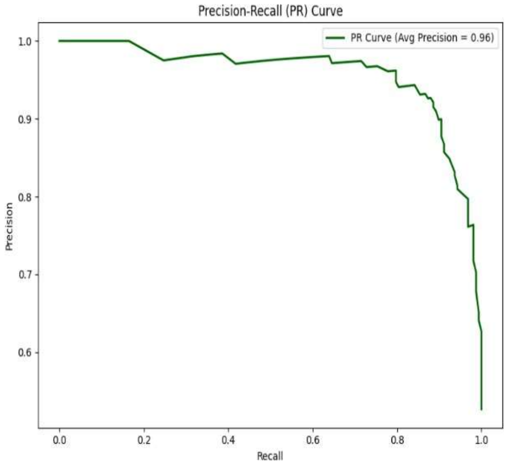

Fig 4: Precision vs. Recall Curve (PR Curve)

This fig 4 curve graphically represents precision on the y-axis and recall on the x-axis, offering insights into how these two metrics trade off with each other. In your scenario, as the threshold for declaring an event as malicious shifts, precision and recall will both change. High precision implies that the model is highly likely to get it right if it classifies an event as malicious, but this may cost in terms of reduced recall and therefore some attacks will be lost. Higher recall, on the other hand, would mean the model is overly sensitive and marking more attacks but at the penalty of reduced precision and possibly identifying more benign behaviors as threats.By examining the PRC, you can calculate the best trade-off for your edge-enabled threat detection system, based on whether you want to minimize false positives (high precision) or capture most threats (high recall). The area under the PRC curve (AUC-PR) is an overall measure of performance, and in cybersecurity applications, a higher AUC-PR means better overall performance at detecting attacks while controlling false alarms.

Table 1: Performance Metrics Before and After Edge Optimization

Metric	Before Optimization	After Optimization
Accuracy (%)	94.2	93.5
Precision (%)	93.8	93.1
Recall (%)	94.0	92.7
F1-Score (%)	93.9	92.9
Inference Latency (ms)	120.0	38.0
Model Size (MB)	255.0	82.0
Epochs	10	10
AUC-ROC Score	0.96	0.95
Average Precision	0.95	0.93

The performance metrics table gives an in-depth comparison of the DistilBERT-based cyber threat detection model prior to and post implementation of edge optimization techniques like weight pruning and quantization. The study points out that while there is a minor loss in classification accuracy—from 94.2% to 93.5%—the overall performance is sound, with marginal drops being observed across precision (93.8% to 93.1%), recall (94.0% to 92.7%), and F1-score (93.9% to 92.9%). These values show that the model remains robust in its predictive ability even after compression for edge deployment. Most importantly, the optimization has brought about a huge boost in computational efficiency. The inference latency has dramatically decreased from 120 milliseconds to as low as 38 milliseconds, and the model size decreased from 255 MB to 82 MB. This dramatic decrease in size improves the model's candidate for deployment onto resource-limited edge and IoT devices where both memory and energy usage are severe issues. Moreover, the score of AUC-ROC very slightly dropped off from 0.96 to 0.95, as well as average precision score that fell from 0.95 to 0.93, showing negligible adverse effect on total model discrimination as well as rank quality.

6 CONCLUSION AND FUTURE WORK

In summary, this research succeeds in demonstrating the feasibility of adopting a compressed model architecture of the transformer model type, i.e., DistilBERT, towards real-time detection of cyber threats in edge computing.

Through leveraging model compression methodologies such as pruning and quantization, and through leveraging dimension reduction methodologies such as PCA, the proposed system effectively succeeds in acquiring high detection rate against restricted computation capabilities of the edge devices. The result is promising, with ongoing improvement in precision and little overfitting, as well as an optimized trade-off between recall and precision. The model is scalable and flexible, meeting the low latency and resource limitations of edge platforms. For further research, further optimization methods can be explored, such as raising the pruning technique with more sophisticated methods, merging reinforcement learning for adaptive threat detection, and expanding the model to accommodate even larger, more varied datasets.

REFERENCE

[1] Mohanarangan, V.D (2020). Improving Security Control in Cloud Computing for Healthcare Environments.Journal of Science and Technology, 5(6).

[2] Aazam, M., Abadal, J., Abarzadeh, M., Abdalla, O. H., Abdelkhalik, O., Aboulian, A., ... & Ashraf, N. (2019). 2019 Index IEEE Transactions on Industrial Informatics Vol. 15. IEEE Transactions on Industrial Informatics, 15(12).

[3] Ganesan, T. (2020). Machine learning-driven AI for financial fraud detection in IoT environments. International Journal of HRM and Organizational Behavior, 8(4).

[4] Garg, S., Singh, A., Batra, S., Kumar, N., & Yang, L. T. (2018). UAV-empowered edge computing environment for cyber-threat detection in smart vehicles. IEEE network, 32(3), 42-51.

[5] Deevi, D. P. (2020). Improving patient data security and privacy in mobile health care: A structure employing WBANs, multi-biometric key creation, and dynamic metadata rebuilding. International Journal of Engineering Research & Science & Technology, 16(4).

[6] Khan, L. U., Yaqoob, I., Tran, N. H., Kazmi, S. A., Dang, T. N., & Hong, C. S. (2020). Edge-computing-enabled smart cities: A comprehensive survey. IEEE Internet of Things journal, 7(10), 10200-10232.

[7] Mohanarangan, V.D. (2020). Assessing Long-Term Serum Sample Viability for Cardiovascular Risk Prediction in Rheumatoid Arthritis. International Journal of Information Technology & Computer Engineering, 8(2), 2347–3657.

[8] Usman, M., Jolfaei, A., & Jan, M. A. (2020). RaSEC: an intelligent framework for reliable and secure multilevel edge computing in industrial environments. IEEE Transactions on Industry Applications, 56(4), 4543-4551.

[9] Koteswararao, D. (2020). Robust Software Testing for Distributed Systems Using Cloud Infrastructure, Automated Fault Injection, and XML Scenarios. International Journal of Information Technology & Computer Engineering, 8(2), ISSN 2347–3657.

[10] Papcun, P., Kajati, E., Cupkova, D., Mocnej, J., Miskuf, M., & Zolotova, I. (2020). Edge-enabled IoT gateway criteria selection and evaluation. Concurrency and Computation: Practice and Experience, 32(13), e5219.

[11] Rajeswaran, A. (2020). Big Data Analytics and Demand-Information Sharing in ECommerce Supply Chains: Mitigating Manufacturer Encroachment and Channel Conflict. International Journal of Applied Science Engineering and Management, 14(2), ISSN2454-9940

[12] Parah, S. A., Kaw, J. A., Bellavista, P., Loan, N. A., Bhat, G. M., Muhammad, K., & de Albuquerque, V. H. C. (2020). Efficient security and authentication for edge-based internet of medical things. IEEE Internet of Things Journal, 8(21), 15652-15662.

[13] Alagarsundaram, P. (2020). Analyzing the covariance matrix approach for DDoS HTTP attack detection in cloud environments. International Journal of Information Technology & Computer Engineering, 8(1).

[14] Xu, X., Zhao, H., Yao, H., & Wang, S. (2020). A blockchain-enabled energy-efficient data collection system for UAV-assisted IoT. IEEE Internet of Things Journal, 8(4), 2431-2443.

[15] Poovendran, A. (2020). Implementing AES Encryption Algorithm to Enhance Data Security in Cloud Computing. International Journal of Information technology & computer engineering, 8(2), I

[16] Usman, M., Jan, M. A., & Jolfaei, A. (2020). SPEED: a deep learning assisted privacy-preserved framework for intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems, 22(7), 4376-4384.

[17] Sreekar, P. (2020). Cost-effective Cloud-Based Big Data Mining with K-means Clustering: An Analysis of Gaussian Data. International Journal of Engineering & Science Research,10(1), 229-249.

[18] Rafique, W., Qi, L., Yaqoob, I., Imran, M., Rasool, R. U., & Dou, W. (2020). Complementing IoT services through software defined networking and edge computing: A comprehensive survey. IEEE Communications Surveys & Tutorials, 22(3), 1761-1804.

[19] Karthikeyan, P. (2020). Real-Time Data Warehousing: Performance Insights of Semi-Stream Joins Using Mongodb. International Journal of Management Research & Review, 10(4), 38-49

[20] Khan, M. N., Rao, A., & Camtepe, S. (2020). Lightweight cryptographic protocols for IoT-constrained devices: A survey. IEEE Internet of Things Journal, 8(6), 4132-4156.

[21] Mohan, R.S. (2020). Data-Driven Insights for Employee Retention: A Predictive Analytics Perspective. International Journal of Management Research & Review, 10(2), 44-59.

[22] Dold, J., & Groopman, J. (2017). The future of geospatial intelligence. Geo-spatial information science, 20(2), 151-162.

[23] Sitaraman, S. R. (2020). Optimizing Healthcare Data Streams Using Real-Time Big Data Analytics and AI Techniques. International Journal of Engineering Research and Science & Technology, 16(3), 9-22.

[24] Shi, F., Ning, H., Huangfu, W., Zhang, F., Wei, D., Hong, T., & Daneshmand, M. (2020). Recent progress on the convergence of the Internet of Things and artificial intelligence. Ieee Network, 34(5), 8-15.

[25] Panga, N. K. R. (2020). Leveraging heuristic sampling and ensemble learning for enhanced insurance big data classification. International Journal of Financial Management (IJFM), 9(1).

[26] Henna, S., & Davy, A. (2020). Distributed and collaborative high-speed inference deep learning for mobile edge with topological dependencies. IEEE Transactions on Cloud Computing, 10(2), 821-834.

[27] Gudivaka, R. L. (2020). Robotic Process Automation meets Cloud Computing: A Framework for Automated Scheduling in Social Robots. International Journal of Business and General Management (IJBGM), 8(4), 49-62.

[28] Mohammed, T., Albeshri, A., Katib, I., & Mehmood, R. (2020). UbiPriSEQ—Deep reinforcement learning to manage privacy, security, energy, and QoS in 5G IoT hetnets. Applied Sciences, 10(20), 7120.

[29] Gudivaka, R. K. (2020). Robotic Process Automation Optimization in Cloud Computing Via Two-Tier MAC and LYAPUNOV Techniques. International Journal of Business and General Management (IJBGM), 9(5), 75-92.

[30] Qadri, Y. A., Nauman, A., Zikria, Y. B., Vasilakos, A. V., & Kim, S. W. (2020). The future of healthcare internet of things: a survey of emerging technologies. IEEE Communications Surveys & Tutorials, 22(2), 1121-1167.

[31] Deevi, D. P. (2020). Artificial neural network enhanced real-time simulation of electric traction systems incorporating electro-thermal inverter models and FEA. International Journal of Engineering and Science Research, 10(3), 36-48.

[32] Jedari, B., Premsankar, G., Illahi, G., Di Francesco, M., Mehrabi, A., & Ylä-Jääski, A. (2020). Video caching, analytics, and delivery at the wireless edge: A survey and future directions. IEEE Communications Surveys & Tutorials, 23(1), 431-471.

[33] Allur, N. S. (2020). Enhanced performance management in mobile networks: A big data framework incorporating DBSCAN speed anomaly detection and CCR efficiency assessment. Journal of Current Science, 8(4).

[34] He, X., Lu, H., Du, M., Mao, Y., & Wang, K. (2020). QoE-based task offloading with deep reinforcement learning in edge-enabled Internet of Vehicles. IEEE Transactions on Intelligent Transportation Systems, 22(4), 2252-2261.

[35] Deevi, D. P. (2020). Real-time malware detection via adaptive gradient support vector regression combined with LSTM and hidden Markov models. Journal of Science and Technology, 5(4).

[36] Ahmad, I., Shahabuddin, S., Sauter, T., Harjula, E., Kumar, T., Meisel, M., ... & Ylianttila, M. (2020). The challenges of artificial intelligence in wireless networks for the Internet of Things: Exploring opportunities for growth. IEEE Industrial Electronics Magazine, 15(1), 16-29.

[37] Dondapati, K. (2020). Integrating neural networks and heuristic methods in test case prioritization: A machine learning perspective. International Journal of Engineering & Science Research, 10(3), 49–56.

[38] Lamb, Z. W., & Agrawal, D. P. (2019). Analysis of mobile edge computing for vehicular networks. Sensors, 19(6), 1303.

[39] Dondapati, K. (2020). Leveraging backpropagation neural networks and generative adversarial networks to enhance channel state information synthesis in millimeter-wave networks. International Journal of Modern Electronics and Communication Engineering, 8(3), 81-90

[40] Shen, M., Xu, K., Du, X., Reed, M. J., Bhuiyan, M. Z. A., Zhang, L., & Mijumbi, R. (2020). Guest Editorial Special Issue on Trust-Oriented Designs of Internet of Things for Smart Cities. IEEE Internet of Things Journal, 7(5), 3897-3900.

[41] Gattupalli, K. (2020). Optimizing 3D printing materials for medical applications using AI, computational tools, and directed energy deposition. International Journal of Modern Electronics and Communication Engineering, 8(3).

[42] Chinamanagonda, S. (2020). Edge Computing: Extending the Cloud to the Edge-Growth in IoT and real-time data processing needs. Advances in Computer Sciences, 3(1).

[43] Allur, N. S. (2020). Big data-driven agricultural supply chain management: Trustworthy scheduling optimization with DSS and MILP techniques. Current Science & Humanities, 8(4), 1–16.

[44] Yu, Y., Tang, X., Wu, J., Kim, B., Song, T., & Han, Z. (2019). Multi-leader–follower game for mec-assisted fusion-based vehicle on-road analysis. IEEE Transactions on Vehicular Technology, 68(11), 11200-11212.

[45] Narla, S., Valivarthi, D. T., & Peddi, S. (2020). Cloud computing with artificial intelligence techniques: GWO-DBN hybrid algorithms for enhanced disease prediction in healthcare systems. Current Science & Humanities, 8(1), 14–30.

[46] Kumar, N., Chaudhry, R., Kaiwartya, O., Kumar, N., & Ahmed, S. H. (2020). Green computing in software defined social internet of vehicles. IEEE Transactions on Intelligent Transportation Systems, 22(6), 3644-3653.

[47] Kethu, S. S. (2020). AI and IoT-driven CRM with cloud computing: Intelligent frameworks and empirical models for banking industry applications. International Journal of Modern Electronics and Communication Engineering (IJMECE), 8(1), 54.

[48] Serhane, O., Yahyaoui, K., Nour, B., & Moungla, H. (2020). A survey of ICN content naming and in-network caching in 5G and beyond networks. IEEE Internet of Things Journal, 8(6), 4081-4104.

[49] Vasamsetty, C. (2020). Clinical decision support systems and advanced data mining techniques for cardiovascular care: Unveiling patterns and trends. International Journal of Modern Electronics and Communication Engineering, 8(2).

[50] Peng, C., Wu, C., Gao, L., Zhang, J., Alvin Yau, K. L., & Ji, Y. (2020). Blockchain for vehicular Internet of Things: Recent advances and open issues. Sensors, 20(18), 5079.

[51] Kadiyala, B. (2020). Multi-swarm adaptive differential evolution and Gaussian walk group search optimization for secured IoT data sharing using supersingular elliptic curve isogeny cryptography,International Journal of Modern Electronics and Communication Engineering,8(3).

[52] Manwal, M. (2019). Smart Healthcare Systems: The Impact of IoT on Medical Diagnostics and Treatment. INFORMATION TECHNOLOGY IN INDUSTRY, 7(3), 68-77.

[53] Valivarthi, D. T. (2020). Blockchain-powered AI-based secure HRM data management: Machine learning-driven predictive control and sparse matrix decomposition techniques. International Journal of Modern Electronics and Communication Engineering.8(4)

[54] Hassan, S. R., Ahmad, I., Ahmad, S., Alfaify, A., & Shafiq, M. (2020). Remote pain monitoring using fog computing for e-healthcare: An efficient architecture. Sensors, 20(22), 6574.

[55] Jadon, R. (2020). Improving AI-driven software solutions with memory-augmented neural networks, hierarchical multi-agent learning, and concept bottleneck models. International Journal of Information Technology and Computer Engineering, 8(2).

[56] Khan, R. A., & Pathan, A. S. K. (2018). The state-of-the-art wireless body area sensor networks: A survey. International Journal of Distributed Sensor Networks, 14(4), 1550147718768994.

[57] Boyapati, S. (2020). Assessing digital finance as a cloud path for income equality: Evidence from urban and rural economies. International Journal of Modern Electronics and Communication Engineering (IJMECE), 8(3).

[58] Li, Y., Zhu, R., Mao, S., & Anjum, A. (2020). Fog-computing-based approximate spatial keyword queries with numeric attributes in IoV. IEEE Internet of Things Journal, 7(5), 4304-4316.

[59] Gaius Yallamelli, A. R. (2020). A cloud-based financial data modeling system using GBDT, ALBERT, and Firefly algorithm optimization for high-dimensional generative topographic mapping. International Journal of Modern Electronics and Communication Engineering8(4).

[60] Garg, S., Singh, A., Batra, S., Kumar, N., & Yang, L. T. (2018). UAV-empowered edge computing environment for cyber-threat detection in smart vehicles. IEEE network, 32(3), 42-51.

[61] Yalla, R. K. M. K., Yallamelli, A. R. G., & Mamidala, V. (2020). Comprehensive approach for mobile data security in cloud computing using RSA algorithm. Journal of Current Science & Humanities, 8(3).

[62] Tian, Z., Luo, C., Qiu, J., Du, X., & Guizani, M. (2019). A distributed deep learning system for web attack detection on edge devices. IEEE Transactions on Industrial Informatics, 16(3), 1963-1971.

[63] Samudrala, V. K. (2020). AI-powered anomaly detection for cross-cloud secure data sharing in multi-cloud healthcare networks. Journal of Current Science & Humanities, 8(2), 11–22.

[64] Xiao, Y., Jia, Y., Liu, C., Cheng, X., Yu, J., & Lv, W. (2019). Edge computing security: State of the art and challenges. Proceedings of the IEEE, 107(8), 1608-1631.

[65] Ayyadurai, R. (2020). Smart surveillance methodology: Utilizing machine learning and AI with blockchain for bitcoin transactions. World Journal of Advanced Engineering Technology and Sciences, 1(1), 110–120.

[66] Puthal, D., Nepal, S., Ranjan, R., & Chen, J. (2016). Threats to networking cloud and edge datacenters in the Internet of Things. IEEE Cloud Computing, 3(3), 64-71.

[67] Chauhan, G. S., & Jadon, R. (2020). AI and ML-powered CAPTCHA and advanced graphical passwords: Integrating the DROP methodology, AES encryption, and neural network-based authentication for enhanced security. World Journal of Advanced Engineering Technology and Sciences, 1(1), 121–132.

[68] Xiao, L., Wan, X., Lu, X., Zhang, Y., & Wu, D. (2018). IoT security techniques based on machine learning: How do IoT devices use AI to enhance security? IEEE Signal Processing Magazine, 35(5), 41-49.

[69] Narla, S. (2020). Transforming smart environments with multi-tier cloud sensing, big data, and 5G technology. International Journal of Computer Science Engineering Techniques, 5(1), 1-10.

[70] Puthal, D., Mohanty, S. P., Bhavake, S. A., Morgan, G., & Ranjan, R. (2019). Fog computing security challenges and future directions [energy and security]. IEEE Consumer Electronics Magazine, 8(3), 92-96.

[71] Alavilli, S. K. (2020). Predicting heart failure with explainable deep learning using advanced temporal convolutional networks. International Journal of Computer Science Engineering Techniques, 5(2).

[72] Alwarafy, A., Al-Thelaya, K. A., Abdallah, M., Schneider, J., & Hamdi, M. (2020). A survey on security and privacy issues in edge-computing-assisted internet of things. IEEE Internet of Things Journal, 8(6), 4004-4022.