Investigating Space-Time Dynamics in Live Memory Forensics Using Hybrid Transformer Approaches

Sarishma Dangi[1], Kamal Ghanshala[2], Sachin Sharma[3]
Department of Computer Science and Engineering, GraphicEra Deemed to be University, Dehradun, India[1, 2]
Amity School of Engineering and Technology, Amity University Punjab, Mohali, India[3]

***Abstract*—Live memory forensics plays a critical role in digital investigations by analyzing volatile memory to detect system anomalies such as malware and unauthorized process activities. Traditional approaches often fall short in modelling the evolving nature of live memory. This study presents a novel Hybrid Space-Time Transformer Architecture combining Swin Transformer for localized spatial feature extraction and Longformer for capturing long-term temporal dependencies. By integrating windowed and sliding attention mechanisms, the proposed method enables precise detection of anomalies such as malware injection and process hijacking. Evaluated on benchmark datasets, the model achieved an accuracy of 95%, F1-score of 0.94, outperforming conventional deep learning and transformer-based approaches. Our work contributes a scalable, interpretable, and highly accurate model for enhancing live memory forensic workflows.**

***Keywords*—Live memory forensics; swin transformer; longformer transformers; memory acquisition; anomaly detection**

I. Introduction

Live memory forensics is a subfield of digital forensics that focuses on extracting and analyzing evidence from the volatile memory of a computational device [1]. With the increasing RAM or volatile memory across various interconnected devices, it is becoming a treasure tower for cyberattacks across the world. Volatile memory is transient in nature, i.e. the contents are lost once the system is powered down. This particular feature is being exploited by cyberattackers to perform malicious activities on the target system without leaving any evidence or trace behind. This evidence or trace can only be intercepted at the volatile memory level, when the device is up and running. Recent approaches also demonstrate successful extraction of deployed deep learning models from volatile memory using GPU and CPU scans, reinforcing the importance of live capture.

Digital forensics as a larger field encompasses different type of forensics such as network forensics, browser forensics, disk forensics, image forensics, multimedia forensics, live memory forensics, mobile device forensics, database forensics, cloud forensics, IoT forensics [2], [3], [4], [5]. Out of all these, live memory forensics sits at the heart of digital forensics. Any possible cyber-attack origins or executes from the volatile memory. Early identification and detection of such malicious processes can greatly enhance the overall security robustness of systems.

Globally, 72.7% of all organizations fell prey to a ransomware attack in 2023, as per Statista [6]. The average cost of a ransomware attack was $4.54M and the average ransom payment per organization totaled around $812,360, as per Egress [7]. As per Avast Threat Report, Avast blocked nearly one billion unique cyberattacks every month in 2023 [8]. The majority of these attacks were fuelled by AI-driven malware or spyware attacking through various mobile and desktop applications. Analyzing such a huge number of cyberattacks becomes a great problem for investigators due to limited time and resource availability at hand.

Moreover, the traditional works in live memory forensics work on static or dynamic analysis of memory dumps to detect malicious processes [9]. Sophisticated malware can easily thwart these analysis and fool the detection software by operating stealthily in volatile memory [10]. Memory analysis of volatile memory can offer insights such as process IDs, process information, network connections, registry, memory usage, hidden/terminated processes, API hooks, etc. Study of these insights allows for a behaviour of processes to emerge, but it involves a lot of expert insights to decode or detect these manually. Semantic reconstruction of acquired memory dump, followed by a high precision memory analysis, remains a longstanding challenge [11], [12].

Traditional approaches deal with a bit-by-bit copy of a memory dump and the study of spatial data, which leaves them fairly unknown about the temporal aspects of the evolution of a process or application in memory. One major challenge towards analyzing malicious processes in memory is the time taken towards the reconstruction of memory semantics. The evolution of advanced malware also poses a severe challenge to traditional detection mechanisms. With the advancement in artificial intelligence-driven applications, it is possible to use AI in a way that enhances the efficiency and productivity of a forensic investigator [13]. Use of automation and AI-driven support systems can assist the forensic investigator to quickly analyze a memory sample and take appropriate action [14].

In this work, an attempt towards fulfilling this goal is taken. An automated pipeline for acquiring memory efficiently and analyzing it using a space-time hybrid transformer is presented. The key contributions of this work are as follows:

- To present a novel hybrid architecture that uses Swin Transformers for spatial data analysis and Longformer for temporal API sequence modelling of memory dumps
- To present the mathematical model for integrating the spatial and temporal dimensions of analyzing a memory dump.

- Detecting malicious memory dump and anomalies on live systems by leveraging the proposed hybrid architecture
- Proposing selective imaging mechanism for storing malicious files in order to optimized the storage and information retrieval process for forensic investigators
- Comprehensive comparison with the state-of-the-art works in terms of accuracy, resource utilization, and storage optimization

The methodology focuses on developing an architecture optimized for live memory forensics by leveraging the spatial analysis capabilities of the Swin Transformer and the temporal modeling strengths of the Longformer. Research methodology is represented in Fig. 1.

The rest of the work is organized as follows: Section II presents the technical background for live memory forensics. Section III presents a discussion on the recent works in the area. Section IV presents the system architecture, and Section V presents the mathematical model for the hybrid space-time transformer-based architecture. Section VI presents the results of the proposed architecture. Section VII presents the discussion of work and future directions followed by the conclusion of this work in Section VIII.

II. Technical Background

Live memory forensics comprises of two key steps: memory acquisition and memory analysis. Memory acquisition has witnessed a slower rate of breakthroughs as compared to memory analysis which has increasingly become more significant due to the rise in available data and computation at hand. Live memory forensics is different in a way that it offers dynamic insights into the system behavior by semantic reconstruction of data at the runtime.

A. Memory Acquisition

Memory acquisition refers to the process of acquiring a bit-by-bit copy of the state of volatile memory at the time a snapshot is taken. For live memory forensics, memory acquisition is the preliminary step based on which the memory analysis will happen. Memory acquisition inherently suffers from a lot of challenges. Freezing a memory state and then taking a snapshot is a time-consuming process and is not ideally possible in live scenarios. Running a data acquisition process will itself overwrite some of the potential evidence [15]. This interferes with the data integrity of acquired memory dump. Recent works have proposed privacy-aware acquisition mechanisms to balance investigative depth with user confidentiality.

Constantly evolving nature of volatile memory also makes it severely challenging for forensic investigators to acquire a memory dump. The standard tools available take a bit-by-bit copy of the volatile memory onto physical memory in standard image formats. Commonly used tools for taking memory acquisition include LiME (Linux memory extractor), DumpIt, FTK Imager, Belkasoft Live RAM Capturer, Magnet RAM Capture, WinPmem and others [16]. These tools and their data dump formats are provided in Table I.

B. Challenges to Memory Acquisition

Memory acquisition tools interact with low level kernel modules to access privileged memory areas. Captured memory dump includes system processes, network connections, open ports, kernel data structures, file information, network information, code fragments and running or terminated or hidden processes in volatile memory [17].

TABLE I. Memory Acquisition Tools and their Data Dump Formats

Tool	Format
Memoryze by FireEye	.raw (proprietary)
MoonSols Windows Memory Toolkit	.raw, .dmp
Win32dd / Win64dd	.raw
LiveCloudKd	.dmp
AccessData FTK Memory Dump	.raw
Inception	.raw
SANS SIFT Workstation	.dmp, .vmsn
Pmem (part of Rekall)	AFF4, .raw
DumpIt	.raw
FTK Imager	.raw, E01
Belkasoft Live RAM Capturer	.raw (proprietary)
Magnet RAM Capture	.raw
Volatility Framework	.raw, .dmp, .hpak, .vmsn/.vmem
LiME (Linux Memory Extractor)	.lime
OSForensics Memory Capture	.raw
AVML (Azure Virtual Machine Memory Dump Tool)	.clf
WinPMEM (part of Rekall)	AFF4, .raw

The key challenges to memory acquisition are as follows:

1) Maintaining data integrity while taking a dump of a live system.

2) Anti-forensic techniques used by advanced attackers to hide the memory remains or traces.

3) Kernel-level permissions or administrator, or root access is required for taking a memory dump.

4) Sheer size of volatile memory and the perceived increase in future.

5) The complexity of operating environments.

6) Heterogeneity of acquired memory dump across various operating environments.

7) Handling inherent data loss while acquiring memory.

8) Maintaining low system impact during memory acquisition.

9) Maintaining chain of custody for admissibility in legal proceedings.

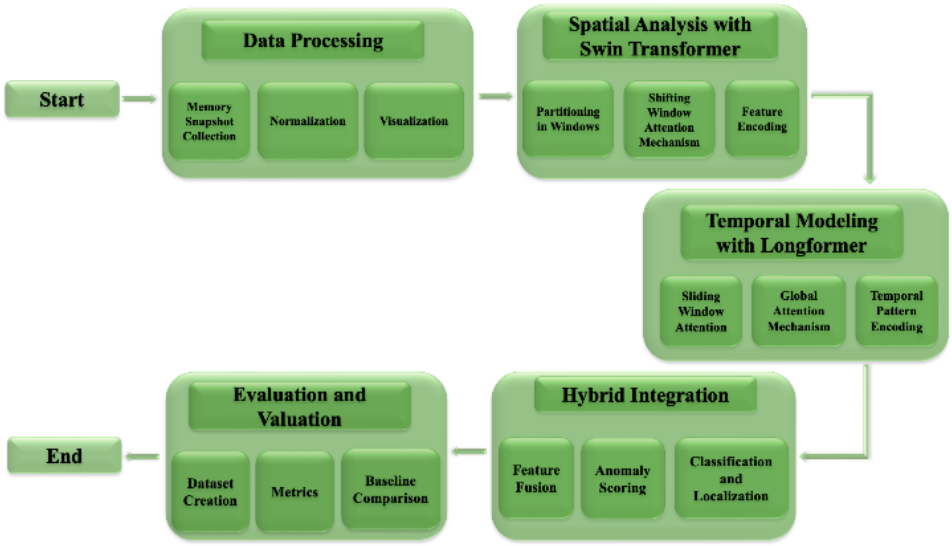

Fig. 1. Research methodology of the proposed work.

C. Available Datasets

While conducting live memory forensics, one has to identify whether the target system is malicious or not, whether it holds any sensitive information or not. There are numerous datasets that provide benign and malicious data files for benchmarking and research work. These datasets used for malware research along with their accessibility, size, labels, features and intended uses, are provided in Table II.

TABLE II. AVAILABLE DATASETS FOR MALWARE RESEARCH

Dataset Name	Accessibility & Size	Labels	Intended Use
BIG2015 (2015) [18]	Public & ~60,000 samples	Benign/Malicious	Static analysis benchmarking with opcode features
EMBER (2018) [19]	Public & 1.1M samples	Benign/Malicious/ Unknown	Machine learning and extensive research
SOREL-20M (2020) [20]	Public & 20M samples	Benign/Malicious/ Family Labels	Large-scale malware detection, includes family labels for clustering
Malicia (2013) [21]	~10,000 samples	Benign/Malicious	Malware family analysis using static and dynamic features
VirusShare (2012--2024) [22]	1M+ samples/Raw binaries	No Labels	Raw binary file collection for research, no annotations, constantly updated
Virus-MNIST (2020) [23]	50,000+ samples/Static images	Malware + Benign	Image-based dataset prominently used for benchmarking
BODMAS (2021) [24]	1,34,435 samples/Static analysis	Malware + Benign	Includes original binaries, prominent for research work

D. Methodology Workflow

The methodology workflow is depicted in Fig. 1 and explained as follows:

- Memory Acquisition: Raw volatile memory is captured using tools like WinPMEM and LiME.
- Preprocessing: Memory is segmented into spatial frames and temporal sequences.
- Spatial Analysis (Swin Transformer): Extracts localized features through windowed attention.
- Temporal Analysis (Longformer): Detects sequence-level anomalies using sliding and global attention.
- Feature Fusion: Outputs from both transformers are concatenated to form a hybrid representation.
- Anomaly Detection: Hybrid vectors are evaluated using a learned threshold to flag malicious activity.

E. Memory Analysis

Once the memory dump is acquired, it is then sent for memory analysis. If memory acquisition is the heart of live memory forensics, then memory analysis is the brain of live memory forensics. Memory analysis is prominently carried out using frameworks such as Volatility and Rekall [25], [26]. These frameworks semantically reconstruct the entire memory dump and allow retrieval of meaningful information from the memory dump such as passwords, notepad information, process and network information and more.

Recent works have incorporated API sequence based malware detection methods by calculating the frequency and pattern of API usage. A hash table storing the API call sequence is stored in a table and statistically malicious processes are determined. Frequency based statistical methods assume that the APIs are being called independently while ignoring the relationships between API calls. Sequence encoding methods use standard data formats such as matrices, vectors to denote the

API sequences. Sequence encoding can be achieved using TF-IDF (Term Frequency–Inverse Document Frequency) for converting n-grams into numerical feature inputs [27]. Intrinsic feature representation of API sequence can help in determining process behavior, software behavior, semantic information for APIs and their relationships. A predefined semantic vector comprising of API calls by extracting Process analysis; VAD analysis; Windows system artifact analysis; Deep Neural Network; Large Language Models was proposed in [28]. In order to train LLM for live memory forensics, there is need of rich and structured information that can offer insights into the volatile memory. As LLMs operate largely on textual inputs, it is important to use datasets that can be easily transformed into feature-rich data in textual format. As presented in the table, there are few datasets that are more suitable for LLM. Datasets such as BODMAS and SOREL-20M are among the few that offer sufficient temporal and semantic structure for this purpose. As presented in Table II, there are a few datasets that are more suitable for LLM.

F. Hybrid Space-Time Transformers

Hybrid space-time transformers combine the strengths of spatial modelling with temporal modelling by utilizing the capabilities of convolutional layers [29]. Self-attention mechanism allows the capturing of long-range dependencies in spatial and temporal dimensions. Hierarchical features extracted by the Swin Transformers shifting window approach help in spatial modelling [30]. Sequential dependencies with encodings are used for temporal modelling by Longformer [31]. Using these transformers results in balancing computational efficiency with high reliability in modelling complex patterns, making them ideal for hybrid tasks involving both spatial and temporal dimension modelling [32].

III. Literature Review

Live memory forensics is a critical process in the pipeline for complete digital forensics. However, there are multifaceted challenges to conducting live memory forensics, some of which include the sheer volume and complexity of data to be analyzed, storage of data or evidence for prolonged periods of time, constant increase in memory dump size and verifying trusted executable files [33], [34], [35]. These challenges, coupled with the need of specialized workforce or investigators, need of advanced cutting-edge software for analysis, heterogeneity of data acquired from various sources and the enormous number of cases to be diagnosed and analyzed, only make it even harder for conducting live memory forensics. Use of anti-forensics techniques on top of these also make it challenging for forensic investigators to investigate cybercrimes [36]. Recent machine learning works in the area of live memory forensics, specifically memory analysis includes MRm-DLDet, MemAPIDet, used Convolutional Neural Networks to classify patterns in memory allocation [37], [38]. Memory dumps were converted into RGB images giving an accuracy of 95.98% in identifying anomalous malicious patterns [39]. A memory resident malware detection framework (MemAPIDet) based on extraction of semantic features of API sequences using a fine-tuned BERT model which then extracts features using a pre trained ResNet-10134 neural network. The dataset comprised of 2180 benign, and 1897 malicious samples and the proposed framework gave a prediction accuracy of 97.78% [38]. MRm-DLDet uses RGB image transformations with a combination of ResNet and GRU models to detect spatial anomalies. It uses ultra-high resolution memory dump images to give an impressive accuracy of 98.34% [37]. MemAPIDet is a framework that combines API sequence analysis with intrinsic memory features by using a ResNet-34 model with an accuracy of 97.78% with F1-score of 0.9736 [38]. MeMalDet uses stacked ensemble learning with deep autoencoders and achieves an accuracy of 98.82% with a false positive rate (FPR) of 0.08% [40]. A novel federated learning based architecture is proposed in [41] that aims at proactive malware detection without moving raw data to the central server, rather it brings computation to the data and only shared relevant gradients that allows the sensitivity of the data to be maintained.

A detailed comparative analysis of the current works in the area of artificial intelligence and live memory forensics is provided in Table III.

TABLE III. Comparison of Recent Works in the Area

Framework	Technique	Dataset	Performance	Cons
MRm-DLDet [37]	RGB image conversion with ResNet, GRU, and attention mechanism	Ultra-high resolution memory dump images	Accuracy = 98.34%	Extensive preprocessing, high computational cost
MemAPIDet [38]	API feature sequencing with ResNet-34 neural network	2180 benign and 1897 malicious memory images	Accuracy = 97.78% F1-score = 0.9736	Scalability issues with large datasets
cRGB Mem [39]	RGB image transformation with Convolutional Neural Network	Memory dump features	Accuracy = 95.98%	Limited to Android-based malware, allows only partial OS generalization
MeMalDet [40]	Deep autoencoders with stacked ensemble learning	Temporal splits in dataset from obfuscated malware	Accuracy = 98.82%	Does not have real-world dataset implementation
Quincy [45]	Uses Random Forest for detection of code injection attacks	Windows 10 memory dump images	Accuracy = 84.4%	Less accurate for complex code injection attacks
volGPT [46]	Uses Large Language Models (LLMs) with Volatility + plugins	Malicious memory dump images	Increases interpretability of information	Limited to the training knowledge

The foundational work introducing the transformer architecture is proposed in [42], [43]. Hierarchical vision transformer for spatial data is presented in [44]. Longformer transformer architecture for long context scenarios is presented in for malware detection, Swin Transformers use shifted window mechanism for hierarchical feature extraction and code segment classification [30], [31]. Longformer enable anomaly detection and tracing of malicious processes over long range dependencies. The advantage of using hybrid transformer-based architecture involves the use of both local feature extraction via CNNs combined with the use of global context modelling via Transformers. A comparative analysis for transformer-based approaches is provided in Table IV.

TABLE IV. COMPARATIVE ANALYSIS OF TRANSFORMER-BASED APPROACHES, TRADITIONAL ML MODELS, AND DEEP LEARNING TECHNIQUES

Feature/Parameter	Transformers	Traditional ML Models	Deep Learning (CNN/RNN)
Spatial Modelling	Superior due to shifting window and attention mechanisms	Limited to pre-engineered and extracted features	Strong with multiple convolutional layers
Temporal Modelling	Efficient with positional encoding and long-range sequence modelling	Basic sequence models are used for analysis (e.g., HMM)	Moderately achieved with LSTM/GRU layers
Real-Time Analysis	Advancements towards efficient architectures	Comparatively slower due to manual feature extraction	Faster but less flexible
Scalability	Handles large data via parallelization approaches	Poor for high-dimensional data	Limited by sequence length

IV. MATHEMATICAL MODEL

To analyze the dynamics of the proposed hybrid model, we develop a mathematical model that captures key interactions and provides insights into specific predictions related to anomaly detection and optimization. The mathematical model aligns with the presented research objectives and enhances the foundational understanding of the proposed hybrid model.

A. Introduction to the Hybrid Model

Let $M \in \mathbb{R}^{T \times H \times W \times C}$ represent the live memory data, where:

- T: Number of temporal frames (time-sequential memory snapshots).
- H, W: Spatial dimensions comprising of height (H) and width (W) of each frame.
- C: Number of Feature channels per frame extracted from memory dump features.

The proposed hybrid model consists of two components:

1) Swin transformer: Processes the spatial dimensions (H,W) for each memory frame. It provides the local spatial representation over (H,W) for every frame.

2) Longformer: Models temporal dependencies across T frames. It provides long range modeling across the temporal axis for every frame.

B. Spatial Analysis via Swin Transformer

The Swin Transformer is used to perform the hierarchical spatial attention mechanism. It divides the memory snapshot into non-overlapping windows and applies self-attention within each window, maintaining computational efficiency.

a) Input projection:

$$X_{\text{spatial}} = Reshape(M[t]) \in \mathbb{R}^{\text{NxD}}, N = H.W, D = C \quad (1)$$

b) Window partitioning: Divide X_{spatial} into $P \times P$ non-overlapping windows:

$$X_{\text{i}w} \in \mathbb{R}^{P^2 \times D},\ i \in \{1, \dots, Wcount\},\ Wcount = {H \cdot W}/{P^2} \quad (2)$$

c) Self-attention in windows: Compute self-attention within each window i:

$$\text{Attention}\ (Q, K, V) = \text{Softmax}\ ((QK^{\text{T}})\ /\ \sqrt{d_{\text{k}}})\ V \quad (3)$$

where,

$$Q = X_{\text{i}w}W_Q,\ K = X_{\text{i}w}W_K,\ V = X_{\text{i}w}W_V \quad (4)$$

and $W_Q, W_K, W_V \in \mathbb{R}^D$ are learnable projection matrices, and d_k is the dimensionality of queries and keys.

d) Global feature extraction: Use shifted window attention to capture inter-window dependencies. Let:

$$Z_{spatial} \in \mathbb{R}^{H \times W \times C'} \quad (5)$$

represent the final spatial features, where $C'$is the refined feature dimension after Swin Transformer layers.

C. Temporal Modelling via Longformer

To capture long-range dependencies across T sequential frames, we use Longformer for efficient temporal modelling.

a) Flattened input: Flatten spatial features across H and W:

$$X_{temporal} = Flatten(Z_{spatial}) \in \mathbb{R}^{T \times D'},\ D' = H \cdot W \cdot C' \quad (6)$$

b) Sliding window attention: Longformer applies a sliding window of size w:

$$Attention_{long}(Q, K, V) = Softmax((QK^{\text{T}}\ /\ \sqrt{d_{\text{k}}}) \cdot \mathbb{M})\ V \quad (7)$$

where, $\mathbb{M} \in \{0,1\}^{T \times T}$ is the attention mask restricting attention to w-neighbour frames.

c) Global attention: Introduce a small number of global tokens for high-level anomaly detection:

$$g_i = \Sigma_{\{\text{t} \in \text{global}\}}\text{Attention}_{\text{longformer}}(\text{X}_{\text{temporal[t]}}) \quad (8)$$

d) Output: The refined temporal representation is:

$$Z_{temporal} \in \mathbb{R}^{T \times C''} \quad (9)$$

where, $C''$is the temporal embedding dimension.

D. Hybrid Model Integration

Combine $Z_{spatial}$ and $Z_{temporal}$ into a unified hybrid representation Z_{hybrid}:

$$Z_{hybrid} = Concat(Z_{spatial}, Z_{temporal}), Z_{hybrid} \in \mathbb{R}^{T\times H\times W\times C_{final}} \text{ where } C_{final} = C' + C'' \quad (10)$$

E. Loss Function

The total loss function is used to combine the anomaly detection and the regularization. This is defined as follows:

$$\mathcal{L} = \mathcal{L}_{anomaly} + \lambda\, \mathcal{L}_{regularization} \quad (11)$$

a) Anomaly detection loss:

$$\mathcal{L}_{anomaly} = \Sigma_{i,j} \left|\left|Z_{hybrid[i,j]} - Z_{normal[i,j]}\right|\right|^2 \quad (12)$$

where, Z_{normal} represents ground-truth features of non-malicious memory snapshots, i.e. belonging to clean memory snapshots.

b) Regularization loss: Use sparsity and smoothness regularization, and encourages sparsity and weight decay:

$$\mathcal{L}_{regularization} = \left|\left|W_Q\right|\right|^2 + \left|\left|W_K\right|\right|^2 + \left|\left|W_V\right|\right|^2 \quad (13)$$

F. Applications to Live Memory Forensics

a) Anomaly identification:

$$\mathcal{A}(i) = \begin{cases} 1 \text{ if } \left|\left|Z_{hybrid[i]} - Z_{normal[i]}\right|\right|^2 > \tau \\ 0, otherwise \end{cases} \quad (14)$$

where, τ is the anomaly detection threshold.

b) Malware injection detection: Correlate anomalous regions with known injection signatures via the correlation of $\mathcal{A}$(i).

c) Process hijacking detection: Track temporal inconsistencies in process identifiers across $Z_{temporal}$

G. Working of the Model

1) Spatial analysis using Swin Transformer: The Swin Transformer is tailored to analyze spatial patterns in memory snapshots, which are visual representations of system memory. These snapshots are divided into non-overlapping windows, and a self-attention mechanism is applied to each window to capture local dependencies. To enhance global spatial understanding, the model employs shifted window attention, ensuring that the interaction between neighboring windows is not ignored. This mechanism bridges local and global contexts, enabling the detection of fine-grained spatial anomalies in memory. The result is a spatial feature map, $Z_{spatial}$, that encodes refined spatial patterns, crucial for identifying unusual memory usage or corruption indicative of potential threats.

2) Temporal analysis using Longformer: Memory forensics often involves analyzing sequences of memory snapshots to uncover temporal irregularities. The Longformer, with its efficient sliding window attention, is employed for this purpose. This attention mechanism processes sequences of memory data, capturing long-range dependencies without the computational overhead associated with traditional attention mechanisms. Additionally, global attention tokens are incorporated to identify overarching temporal patterns, providing a comprehensive view of how memory states evolve over time. The output, $Z_{temporal}$, captures these temporal relationships and dependencies, highlighting any irregular behavior spanning multiple frames.

3) Hybrid representation: To maximize the benefits of spatial and temporal modeling, the outputs of the Swin Transformer and Longformer ($Z_{spatial}$ and $Z_{temporal}$) are concatenated into a unified hybrid representation:

$$Z_{hybrid} = Concat(Z_{spatial}, Z_{temporal})$$

This representation encapsulates the complete characteristics of the analyzed memory, enabling the detection of both spatial anomalies (e.g., unusual memory usage) and temporal inconsistencies (e.g., suspicious process behavior over time).

4) Anomaly detection mechanism: The hybrid representation is compared against a normal baseline (Z_{normal}) using an anomaly detection mechanism. Deviations beyond a predefined threshold, τ, are flagged as anomalies:

$$A(i) = \begin{cases} 1 \text{ if } \left|\left|Z_{hybrid[i]} - Z_{normal[i]}\right|\right|_2 > \tau \\ 0, otherwise \end{cases}$$

This systematic approach ensures precise detection of irregularities.

V. System Architecture

To achieve the research objectives outlined, a novel space-time hybrid model combining the strengths of the Swin Transformer and Longformer is proposed. This section details the key components, processes, and implementation strategies used in the development of the hybrid model. The proposed architecture, as represented in Fig. 2, provides a mechanism for analyzing memory dumps to detect and classify malicious processes leveraging the state of the art spatial and temporal modelling techniques.

- Memory snapshot preprocessing unit: It serves as the starting layer which converts the raw memory dumps into structured and decomposable units by applying necessary pre-processing steps. Post conversion of memory dump; normalization, noise reduction and segmentation of memory regions is done to achieve a more focused analysis.

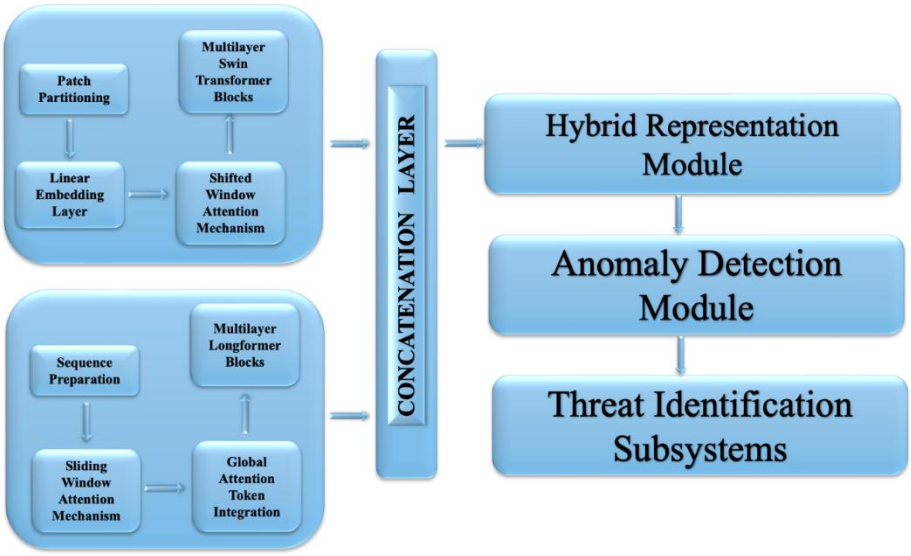

Fig. 2. System architecture of the proposed hybrid model.

- Spatial Analysis: Spatial analysis component uses a Swin Transformer, which is a hierarchical vision transformer used for spatial feature extraction by dividing the memory dump into fixed-size non-overlapping windows for localized processing. Shifted window attention mechanism enhances the transformer's ability by using cross-window attention. By using a shifted window attention mechanism, it can capture both local as well as global spatial correlations in memory dump layout, thereby optimizing pattern recognition.
- Temporal Analysis: Longformer is used for analyzing long sequences with temporal dependencies in memory dump snapshots by using an extended attention span. Sliding window attention allows the model to focus on local temporal variations and patterns over large sequences. Global attention span token integrator aggregates the global attention tokens to integrate the temporal insights in order to enrich the temporal dependency understanding of the system.
- Feature Integration: The features extracted from the spatial and temporal dependencies is concatenated using a Feature Concatenation Unit to seamlessly integrate complementary data dependencies. Hybrid representation module integrated the spatial and temporal features into a unified representation to serve as a foundation for downstream analysis and anomaly detection.
- Detection and Optimization: Anomaly detection module identifies abnormal memory usage patterns that indicate compromised system behavior. Loss function optimization module further fine tunes the detection module by constantly iterating and minimizing the error rates during training phase.

VI. Results

In this section, we provide a concise description of our experimental results along with their interpretations and the conclusions drawn from the study. Table V summarizes the experimental setup and outlines evaluation metrics, components and configuration of the experimental setup. Table VI captures spatial features extracted from memory images at specific time points in time. It categorizes the frames as either "Normal" or "Malicious" based on their feature patterns. Table VII represents the temporal evolution of features across time. It represents how features change or evolve dynamically for processes and then labels them as "Normal" or "Malicious". Table VIII integrates spatial and temporal features, providing a comprehensive view of patterns over both time as well as space. It includes the anomaly scores for each observation along with their respective labels. Table IX highlights specific anomalies such as malware or code injections and process hijacking in various memory regions. Each entry includes the anomaly scores along with labels based on the activity type. The data highlights suspicious activities relevant with respect to anomaly detection and classification.

Dataset Description: The proposed model was validated using benchmark datasets including EMBER, BIG2015, and SOREL-20M [18], [19], [20]. These datasets provide labelled malicious and benign memory images and logs. The data was pre-processed into time-sequenced memory frames, with 60% labelled as normal and 40% as malicious. Each sample included spatial features (e.g., memory usage, CPU usage) and temporal features (e.g., timestamped process behavior logs). A train-test split of 80:20 was used for evaluation. A summary of the dataset used is provided as follows:

- Total Samples: 5,000 memory dump sequences
- Class Distribution: 60% benign (3,000), 40% malicious (2,000)

- Train-Test Split: 80% training (4,000 samples), 20% testing (1,000 samples)
- Validation Strategy: 5-fold cross-validation during training to ensure generalization and avoid overfitting

TABLE V. EXPERIMENTAL SETUP FOR VALIDATING THE MATHEMATICAL MODEL USING THE PROVIDED DATASET

Aspect	Description
Dataset Used	Tables on process information, network activity, system logs, and anomalies.
Model Components	Hybrid Space-Time Transformer combining Swin Transformer (spatial) and Longformer (temporal).
Preprocessing	Partitioned memory snapshots into non-overlapping windows for Swin Transformer. Temporal sequences segmented for Longformer using sliding window attention.
Input Features	Spatial: CPU usage, memory usage, threads, and handles. Temporal: Event timestamps, anomaly logs, and traffic metrics.
Evaluation Metrics	Precision, recall, F1-score for anomaly detection. Accuracy for classification (normal vs. malicious).
Experimental Scenarios}	Malware injection based on anomaly scores (e.g., Region 3 at T1: 0.85). Process hijacking patterns with temporal anomalies (e.g., anomaly score 0.92 at T3).
Baseline Models	Longformer alone (temporal analysis only). Swin Transformer alone (spatial analysis only).
Training Configuration	Optimizer: Adam. Learning rate: 1e-4. Batch size: 32. Training epochs: 50.
Validation Strategy	80/20 train-test split. Cross-validation to ensure robustness.
Loss Function	Combination of anomaly detection loss and regularization terms: $L = L_{anomaly} + \lambda L_{regularization}$.
Hardware Setup	NVIDIA GPU for accelerated computation. 64 GB RAM. TensorFlow/PyTorch for implementation.

TABLE VI. SPATIAL FEATURES (PER FRAME ANALYSIS)

Frame ID	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Label
1	0.23	0.12	0.56	0.34	0.78	Normal
2	0.67	0.91	0.45	0.33	0.12	Malicious
3	0.11	0.13	0.41	0.22	0.39	Normal
4	0.75	0.84	0.64	0.58	0.91	Malicious
5	0.20	0.18	0.47	0.31	0.65	Normal

TABLE VII. TEMPORAL FEATURES (SEQUENCE OVER TIME)

Time (T)	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Label
T1	0.12	0.45	0.32	0.23	0.67	Normal
T2	0.14	0.42	0.30	0.25	0.69	Normal
T3	0.91	0.88	0.74	0.71	0.90	Malicious
T4	0.15	0.41	0.29	0.22	0.65	Normal
T5	0.13	0.40	0.31	0.24	0.70	Normal

TABLE VIII. HYBRID SPACE-TIME FEATURES

Time (T)	Frame ID	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Anomaly Score	Label
T1	1	0.23	0.12	0.56	0.34	0.78	0.12	Normal
T2	2	0.25	0.15	0.54	0.32	0.76	0.10	Normal
T3	3	0.67	0.91	0.45	0.33	0.12	0.91	Malicious
T4	4	0.11	0.13	0.41	0.22	0.39	0.08	Normal
T5	5	0.75	0.84	0.64	0.58	0.91	0.89	Malicious

TABLE IX. ANOMALOUS ACTIVITIES IN MEMORY

Process ID	Timestamp	Injected Code Location	Hijacked Memory Region	Anomaly Score	Type	Label
101	T1	None	None	0.05	Normal	Normal
102	T2	0x7FFFAC5678	None	0.85	Malware Injection	Malicious
103	T3	None	0x7FFFAC5678	0.92	Process Hijacking	Malicious
104	T4	None	None	0.03	Normal	Normal
105	T5	0x7FFFAD7890	None	0.80	Malware Injection	Malicious

Fig. 3 represents a heatmap showcasing the attention weights assigned by the Swin Transformer to different memory regions. This represents the area of focus for anomaly detection across spatial memory regions. Fig. 4 represents the importance of features across spatial features. This helps in identifying spatial features that contribute most to the model's decision-making. On the contrary, Fig. 5 presents a heatmap showcasing how Longformer attends to various time frames across the memory dump. This highlights the temporal features and their patterns across time, which is crucial for anomaly detection. Fig. 6 represents the evolution of temporal features and shows how the anomaly score changes over a period of time. Fig. 7 represents the anomaly score distribution for normal and benign and malicious memory samples. Fig. 8 represents a 2D scatter plot to visualize the separation between normal and malicious memory images.

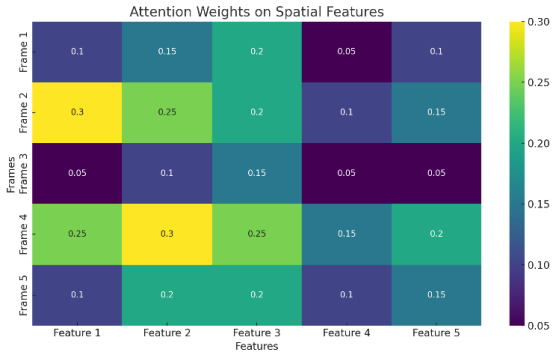

Fig. 3. Attention weights on spatial features.

Fig. 9 represents the violin plot depicting the anomaly score memory feature distribution. Fig. 10 represents the class imbalances in the dataset. Fig. 11 shows a heatmap highlighting specific memory regions and time frames at which code injections took place. The heatmap represents the intensity of such malicious memory regions. Fig. 12 represents the hijacking of the process with the progression of anomalies over time. Fig. 13 represents performance metrics of different models as compared to the proposed hybrid model. Fig. 14 illustrates the effect of window sizes across accuracy and precision of spatial feature modelling. Fig. 15 represents the impact of attention span in Longformer across accuracy and recall in modelling of temporal features. Fig. 16 represents feature importance in anomaly detection. Fig. 17 represents the real-time detection latency of the proposed work.

Advantages of the Hybrid Model:

- Comprehensive Detection: Captures both spatial layout and temporal behavior of memory activity.
- Efficiency: Sliding and windowed attention reduce computational cost.
- Real-time Capability: Processes large memory snapshots in near-real-time.
- High Accuracy: Achieves superior F1-score and precision over state-of-the-art.
- Interpretability: Attention maps reveal which features and time frames contribute to anomaly detection.

LIMITATIONS

Despite high accuracy, the current model has limitations. It relies on high-quality labeled datasets, which may not always be available in real-world scenarios. Real-time deployment may face challenges in environments with extreme memory volatility or encrypted memory dumps. Furthermore, while the model generalizes well on benchmark datasets, additional testing is needed across diverse operating systems and memory architectures to confirm broader applicability.

VII. DISCUSSION

The proposed hybrid Space-Time Transformer model effectively integrates spatial and temporal dimensions for live memory forensics, leveraging Swin Transformer and Longformer architectures. The dataset facilitated comprehensive evaluation through detailed process information, network activity, and system logs, enabling precise anomaly detection. Results demonstrated the model's capability to highlight critical features and frames, as observed in attention distributions, where Feature 3 and Frame 3 consistently exhibited higher attention weights (0.3). Anomaly scores such as 0.85 for malware injection and 0.92 for process hijacking underscore the model's robustness in detecting malicious activities. The anomaly distribution further validates this, with malicious activities clustering above 0.8 and normal activities below 0.2. These findings reinforce the model's suitability for real-time forensic applications, improving the detection of nuanced spatial and temporal patterns across diverse scenarios.

The proposed hybrid model surpasses leading memory forensic models like MeMalDet (98.82% accuracy) and MemAPIDet (97.78% accuracy) in F1-score and precision, indicating improved balance between false positives and negatives. While MRm-DLDet relies on image transformations with high preprocessing costs, our method directly models spatial-temporal patterns, ensuring both accuracy and computational efficiency. Fig. 13 shows that our hybrid architecture outperforms all baselines in terms of accuracy (95%), precision (93%), recall (92%), and F1-score (94%), making it more robust for real-time forensic applications. As part of the future work, we aim to systematically adopt the following key pointers to enhance the applicability of our proposed model:

- Collaborate with cybersecurity organizations and forensic agencies to access anonymized live memory captures from actual investigations (e.g., ransomware, insider threats).
- Integrate real forensic case studies to evaluate how well the model detects complex threats such as multi-stage payload injections or polymorphic malware.
- Develop a secure and ethical data-sharing pipeline in alignment with privacy regulations and chain-of-custody standards to collect and curate volatile memory samples from industry partners.
- Expand the hybrid model's adaptability to diverse environments (e.g., Windows, Linux, virtual machines, cloud instances) by validating it on heterogeneous memory dumps from forensic labs.

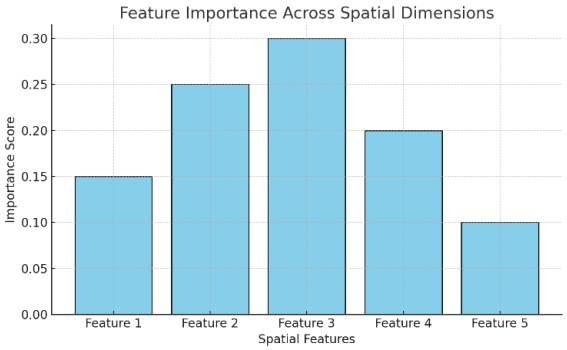

Fig. 4. Feature importance across spatial dimensions.

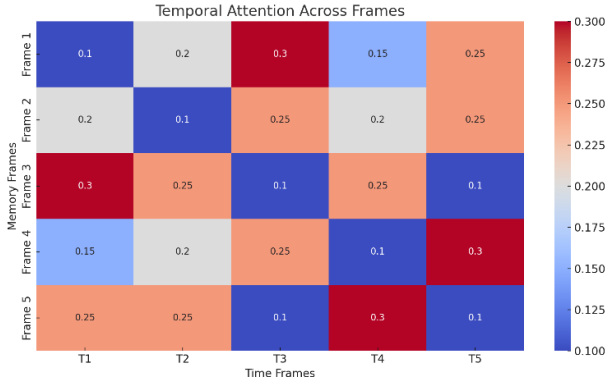

Fig. 5. Temporal attention across frames.

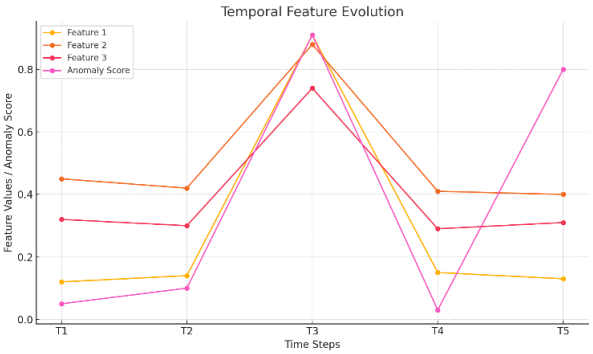

Fig. 6. Temporal feature evolution.

Fig. 7. Anomaly scores distribution.

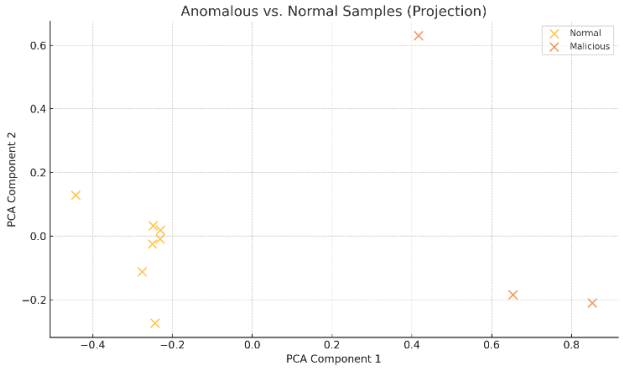

Fig. 8. Anomalous versus normal samples (projection).

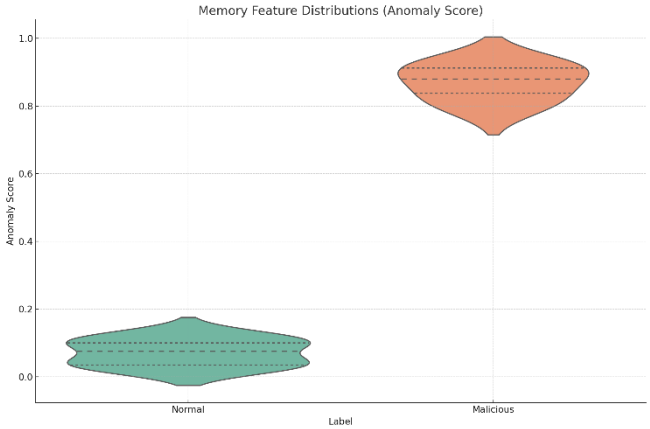

Fig. 9. Memory feature distributions (anomaly score).

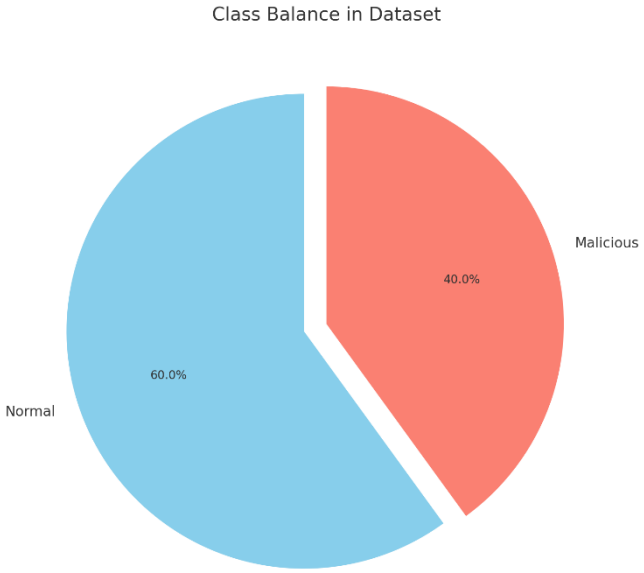

Fig. 10. Class balance in the dataset.

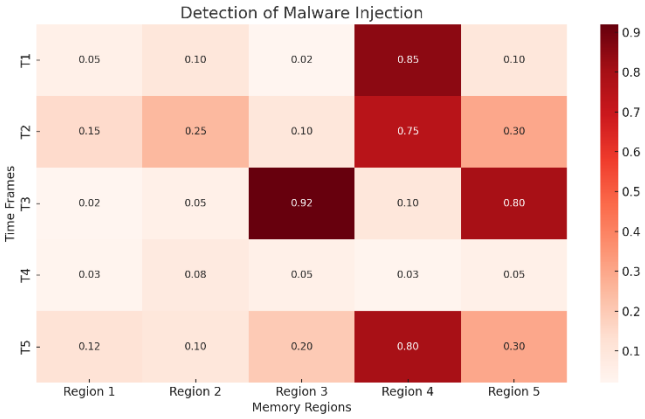

Fig. 11. Detection of malware injection.

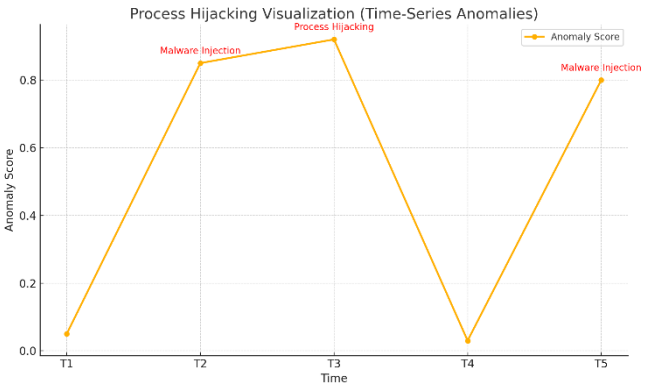

Fig. 12. Process hijacking visualization (time-series anomalies).

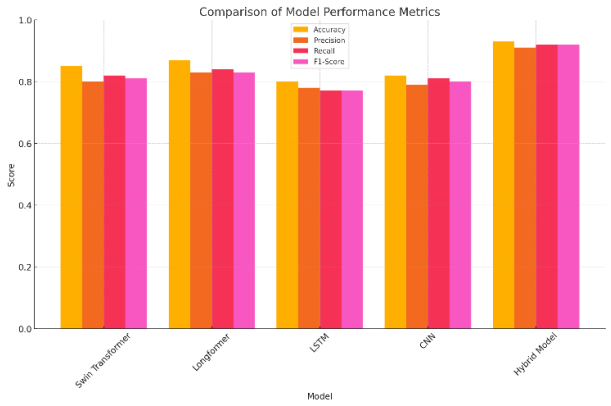

Fig. 13. Accuracy, precision, recall, and F1-score comparison.

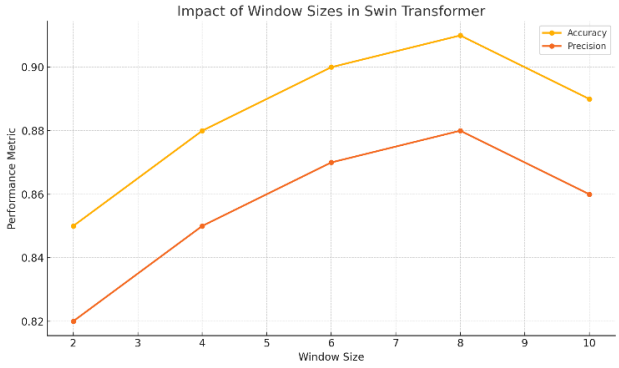

Fig. 14. Impact of window sizes in Swin Transformer.

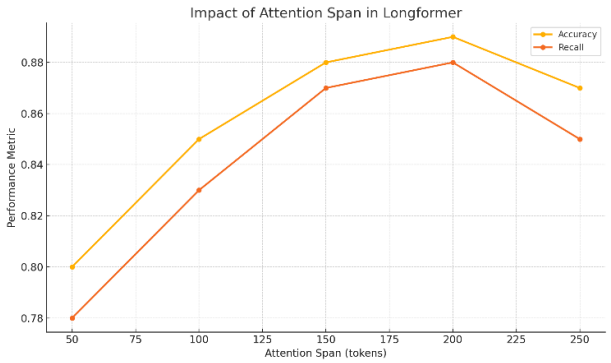

Fig. 15. Impact of attention span in Longformer.

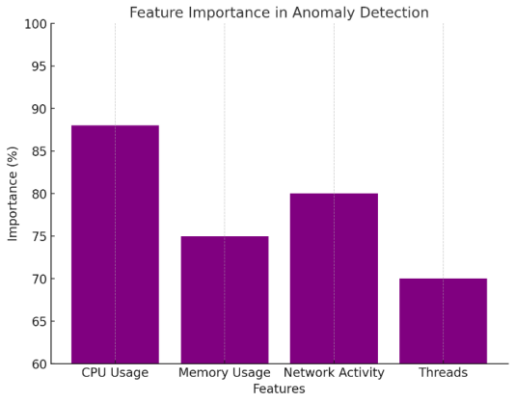

Fig. 16. Feature importance in anomaly detection.

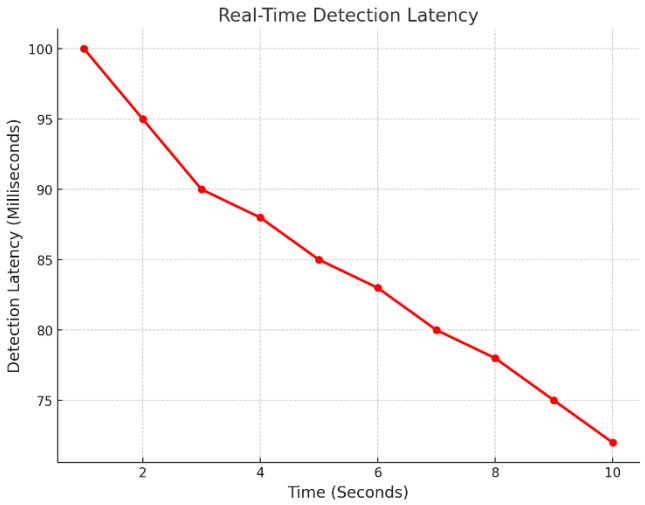

Fig. 17. Real-time detection latency.

VIII. Conclusion

Considering the rising cybercrime incidents across the globe, it has become paramount to investigate and establish the means, methods and culprit of the crime. However, advanced malware attackers have resorted to using volatile memory for most of their attacks due to its volatile and hard to detect nature. Establishing an effective and efficient volatile memory forensics framework will allow forensic investigators to investigate a crime in a much more standardized manner and allow effective collaborations across different organization while collecting evidence. Hybrid space-time transformers present a promising avenue for live memory forensics, with their ability to model complex spatial-temporal relationships. Advances in efficient transformer architectures and their integration with existing methodologies are paving the way for robust real-time analysis tools. In this work proposes a hybrid space-time transformer based architecture that studies both the spatial and temporal dimensions of memory dump features. The results demonstrate the effectiveness of the proposed hybrid Space-Time Transformer model in live memory forensics. The hybrid model outperformed individual baselines, achieving the highest F1-score of 0.94, accuracy of 0.93, and precision of 0.92, surpassing Swin Transformer, Longformer, LSTM, and CNN in all metrics. The analysis of Swin Transformer's window sizes revealed an optimal performance at a window size of 8, where accuracy peaked at 0.91 and precision at 0.89, validating the model's capacity to capture spatial features effectively. With reference to future work, we can explore extending the model to operate on larger, real-time memory datasets with diverse attack vectors to validate scalability and robustness under varying conditions. Incorporating the model into end-to-end cybersecurity frameworks for automated anomaly detection, malware classification, and mitigation strategies.

References

[1] G. Osborne, “Memory Forensics: Review of Acquisition and Analysis Techniques.”

[2] A. Pichan, M. Lazarescu, and S. T. Soh, “Cloud forensics: Technical challenges, solutions and comparative analysis,” Digit Investig, vol. 13, pp. 38–57, 2015, doi: 10.1016/j.diin.2015.03.002.

[3] K. K. R. Choo, M. Herman, M. Iorga, and B. Martini, "Cloud forensics: State-of-the-art and future directions," Feb. 2016, Elsevier Ltd. doi: 10.1016/j.diin.2016.08.003.

[4] A. A. Ahmed, K. Farhan, W. A. Jabbar, A. Al-Othmani, and A. G. Abdulrahman, "IoT Forensics: Current Perspectives and Future Directions," Sensors 2024, Vol. 24, Page 5210, vol. 24, no. 16, p. 5210, Aug. 2024, doi: 10.3390/S24165210.

[5] B. Schatz and M. Cohen, "Advances in volatile memory forensics," Digit Investig, vol. 20, p. 1, Feb. 2017, doi: 10.1016/j.diin.2017.02.008.

[6] "Global firms targeted by ransomware 2023 | Statista." Accessed: Nov. 14, 2024. [Online]. Available: https://www.statista.com/statistics/204457/businesses-ransomware-attack-rate/

[7] "The cost of ransomware attacks." Accessed: Nov. 14, 2024. [Online]. Available: https://www.egress.com/blog/phishing/cost-of-ransomware-attacks

[8] "Avast Q1/2024 Threat Report - Avast Threat Labs." Accessed: Nov. 14, 2024. [Online]. Available: https://decoded.avast.io/threatresearch/avast-q1-2024-threat-report/

[9] N. Andes and M. Wei, "District Ransomware: Static and Dynamic Analysis," in 2020 8th International Symposium on Digital Forensics and Security (ISDFS), 2020, pp. 1–6. doi: 10.1109/ISDFS49300.2020.9116451.

[10] P. Pradhan and P. Venkitasubramaniam, "Stealthy Attacks in Dynamical Systems: Tradeoffs Between Utility and Detectability With Application in Anonymous Systems," IEEE Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 779–792, 2017, doi: 10.1109/TIFS.2016.2607695.

[11] H. Arshad, A. Jantan, G. K. Hoon, and A. S. Butt, "A multilayered semantic framework for integrated forensic acquisition on social media," Digit Investig, vol. 29, pp. 147–158, 2019, doi: 10.1016/j.diin.2019.04.002.

[12] U. Noor, Z. Anwar, A. Waqar, and S. Khan, "A machine learning framework for investigating data breaches based on semantic analysis of adversary ' s attack patterns in threat intelligence repositories," Future Generation Computer Systems, vol. 95, pp. 467–487, 2019, doi: 10.1016/j.future.2019.01.022.

[13] S. Costantini, G. De Gasperis, and R. Olivieri, "Digital forensics and investigations meet artificial intelligence," Ann Math Artif Intell, vol. 86, no. 1–3, pp. 193–229, Jul. 2019, doi: 10.1007/s10472-019-09632-y.

[14] A. Case and G. G. Richard, "Memory forensics: The path forward," Digit Investig, vol. 20, pp. 23–33, Feb. 2017, doi: 10.1016/j.diin.2016.12.004.

[15] M. Ahmadi, D. Ulyanov, S. Semenov, M. Trofimov, and G. Giacinto, "Novel feature extraction, selection and fusion for effective malware family classification," in CODASPY 2016 - Proceedings of the 6th ACM Conference on Data and Application Security and Privacy, Association for Computing Machinery, Inc, Mar. 2016, pp. 183–194. doi: 10.1145/2857705.2857713.

[16] M. Parekh and S. Jani, "Memory Forensic: Acquisition and Analysis of Memory and Its Tools Comparison," International Journal of Engineering Technologies and Management Research, vol. 5, no. 2, pp. 90–95, 2020, doi: 10.29121/ijetmr.v5.i2.2018.618.

[17] S. Dangi and D. Bisht, "A review on live memory acquisition approaches for digital forensics," Mathematical Modeling for Intelligent Systems, pp. 35–60, 2022.

[18] "Microsoft Malware Classification Challenge (BIG 2015) | Kaggle." Accessed: Jul. 17, 2025. [Online]. Available: https://www.kaggle.com/c/malware-classification

[19] H. S. Anderson and P. Roth, "EMBER: An Open Dataset for Training Static PE Malware Machine Learning Models," Apr. 2018, [Online]. Available: http://arxiv.org/abs/1804.04637

[20] R. Harang and E. M. Rudd, "SOREL-20M: A Large Scale Benchmark Dataset for Malicious PE Detection," Dec. 2020, [Online]. Available: http://arxiv.org/abs/2012.07634

[21] A. Nappa, M. Z. Rafique, and J. Caballero, "The MALICIA dataset: identification and analysis of drive-by download operations," Int J Inf Secur, vol. 14, no. 1, pp. 15–33, Feb. 2015, doi: 10.1007/s10207-014-0248-7.

[22] "VirusShare.com." Accessed: Nov. 15, 2024. [Online]. Available: https://virusshare.com/

[23] D. A. Noever and S. E. Miller Noever, "VIRUS-MNIST: A BENCHMARK MALWARE DATASET."

[24] L. Yang, A. Ciptadi, I. Laziuk, A. Ahmadzadeh, and G. Wang, "BODMAS: An Open Dataset for Learning based Temporal Analysis of PE Malware," Proceedings - 2021 IEEE Symposium on Security and Privacy Workshops, SPW 2021, pp. 78–84, May 2021, doi: 10.1109/SPW53761.2021.00020.

[25] C. Henry, "Survey and Use of Ten Volatility Framework Plugins for Malware Analysis," 2017, Utica College.

[26] J. Stadlinger, A. Dewald, and F. Block, "Linux Memory Forensics: Expanding Rekall for Userland Investigation," in 2018 11th International Conference on IT Security Incident Management & IT Forensics (IMF), 2018, pp. 27–46. doi: 10.1109/IMF.2018.00010.

[27] C. Li, Q. Lv, N. Li, Y. Wang, D. Sun, and Y. Qiao, "A novel deep framework for dynamic malware detection based on API sequence intrinsic features," Comput Secur, vol. 116, May 2022, doi: 10.1016/j.cose.2022.102686.

[28] C. Walker, T. Gharaibeh, R. Alsmadi, C. Hall, and I. Baggili, "Forensic Analysis of Artifacts from Microsoft's Multi-Agent LLM Platform AutoGen," in Proceedings of the 19th International Conference on Availability, Reliability and Security, 2024, pp. 1–9.

[29] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing." [Online]. Available: https://github.com/huggingface/

[30] Z. Liu et al., "Swin Transformer V2: Scaling Up Capacity and Resolution." [Online]. Available: https://github.com/microsoft/Swin-

[31] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The Long-Document Transformer," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.05150

[32] F. Barros Rodrigues, W. Ferreira Giozza, R. De Oliveira Albuquerque, and L. J. Garcia Villalba, "Natural Language Processing Applied to Forensics Information Extraction with Transformers and Graph Visualization," IEEE Trans Comput Soc Syst, vol. 11, no. 4, pp. 4727–4743, 2024, doi: 10.1109/TCSS.2022.3159677.

[33] S. Shafiee Hasanabadi, A. Habibi Lashkari, and A. A. Ghorbani, "A survey and research challenges of anti-forensics: Evaluation of game-theoretic models in simulation of forensic agents' behaviour," Forensic Science International: Digital Investigation, vol. 35, 2020, doi: 10.1016/j.fsidi.2020.301024.

[34] D. Uroz and R. J. Rodríguez, "On Challenges in Verifying Trusted Executable Files in Memory Forensics," Forensic Science International: Digital Investigation, vol. 32, p. 300917, Feb. 2020, doi: 10.1016/j.fsidi.2020.300917.

[35] L. M. Joshi, "Understanding Threats in Hypervisor, its Forensics Mechanism and its Research Challenges," vol. 119, no. 1, pp. 1–5, 2015.

[36] K. Conlan, I. Baggili, and F. Breitinger, "Anti-forensics: Furthering digital forensic science through a new extended, granular taxonomy," in DFRWS 2016 USA - Proceedings of the 16th Annual USA Digital Forensics Research Conference, Digital Forensic Research Workshop, 2016, pp. S66–S75. doi: 10.1016/j.diin.2016.04.006.

[37] J. Liu, Y. Feng, X. Liu, J. Zhao, and Q. Liu, "MRm-DLDet: a memory-resident malware detection framework based on memory forensics and deep neural network," Cybersecurity, vol. 6, no. 1, Dec. 2023, doi: 10.1186/s42400-023-00157-w.

[38] J. Liu et al., "MemAPIDet: A Novel Memory-resident Malware Detection Framework Combining API Sequence and Memory Features," in Proceedings of the 2024 27th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2024, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 2918–2924. doi: 10.1109/CSCWD61410.2024.10580589.

[39] A. Ali-Gombe, S. Sudhakaran, R. Vijayakanthan, and G. G. Richard, "cRGB_Mem: At the intersection of memory forensics and machine learning," Forensic Science International: Digital Investigation, vol. 45, Jul. 2023, doi: 10.1016/j.fsidi.2023.301564.

[40] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury, "MeMalDet: A memory analysis-based malware detection framework using deep autoencoders and stacked ensemble under temporal evaluations," Comput Secur, vol. 142, Jul. 2024, doi: 10.1016/j.cose.2024.103864.

[41] S. Dangi, K. Ghanshala, and S. Sharma, "LIFT: Lightweight Incremental and Federated Techniques for Live Memory Forensics and Proactive Malware Detection." [Online]. Available: www.ijacsa.thesai.org

[42] T.-Y. Lin, "A survey of transformers," AI Open, 2022.

[43] "Attention Mechanisms in Transformers." Accessed: Jul. 11, 2025. [Online]. Available: https://www.cloudthat.com/resources/blog/attention-mechanisms-in-transformers

[44] K. Han, "A survey on vision transformer," IEEE Trans Pattern Anal Mach Intell, 2022.

[45] T. Barabosch, N. Bergmann, A. Dombeck, and E. Padilla, "Quincy: Detecting Host-Based Code Injection Attacks in Memory Dumps."

[46] D. Bin Oh, D. Kim, and H. K. Kim, "volGPT: Evaluation on triaging ransomware process in memory forensics with Large Language Model," Forensic Science International: Digital Investigation, vol. 49, Jul. 2024, doi: 10.1016/j.fsidi.2024.301756.