Hybrid Deep Learning Model for Multiple Cache Side Channel Attacks Detection: A Comparative Analysis

Tejal Joshi

Dept. of Computer Science & Engineering
School of Computational Sciences
COEP Technological University (COEP TECH)
Pune, India
joshits21.comp@coeptech.ac.in

Anvi Jamkhande

Dept. of Computer Science & Engineering
School of Computational Sciences
COEP Technological University (COEP TECH)
Pune, India
jamkhandeaa21.comp@coeptech.ac.in

Aarya Kawalay

Dept. of Computer Science & Engineering
School of Computational Sciences
COEP Technological University (COEP TECH)
Pune, India
aaryark21.comp@coeptech.ac.in

Amit Joshi

Senior Member, IEEE

Dept. of Computer Science & Engineering
School of Computational Sciences

COEP Technological University (COEP TECH)

Pune, India
adj.comp@coeptech.ac.in

Abstract—Cache side-channel attacks have emerged as a sophisticated and persistent threat, capable of extracting sensitive user information by exploiting vulnerabilities in modern processors. These attacks leverage the inherent weaknesses in shared computational resources, especially the last-level cache. They infer patterns in data access and execution flows, often bypassing traditional security defenses. These are particularly dangerous because they require no physical access to the victim's device, making remote attacks feasible. This study focuses on a specific class of these threats— fingerprinting attacks —where an adversary can use cache-side channels to monitor and analyze the behavior of co-located processes, potentially revealing confidential information such as encryption keys or user activity patterns. A comprehensive threat model illustrates how an attacker, sharing computational resources with a target system, can exploit these side-channels. This exploitation allows the attacker to learn patterns in data access, potentially compromising sensitive information. To mitigate such risks, a hybrid deep learning model is proposed to detect cache side-channel attacks. Its performance is compared with five widely-used Deep Learning models like Multi Layer Perceptron, Convolutional Neural Network, Simple Recurrent Neural Network, Long Short Term Memory, and Gated Recurrent Unit, evaluating each model's resilience to these sophisticated attacks. The experimental results suggest that the hybrid model has achieved a detection rate of up to 99.96%. The findings demonstrate the limitations of existing models and emphasize the need

for enhanced defensive mechanisms, shedding light on future developments for securing sensitive data against evolving side-channel threats.

Index Terms—Deep Learning, Cache Side-Channel Attacks, Hardware Performance Counters, Hybrid Model, Secured Architecture, Time Series

I. INTRODUCTION

Cache side-channel attacks (CSCA) exploit the shared hardware resources, particularly CPU caches, to extract sensitive information without needing direct access to the target system [1] [2] [3]. By monitoring the cache's behavior, such as timing variations and cache hits or misses, attackers can infer critical data of the users. These attacks pose significant threats across various environments, from cloud computing to personal devices [4] [5]. Despite advancements in security, these attacks remain prevalent due to the evolving techniques and complexities of modern systems, making them a persistent concern. Implementing secure architecture is essential to safeguard these environments, as it helps in minimizing vulnerabilities that attackers can exploit through shared hardware resources.

Various kinds of CSCAs, like PRIME+PROBE & FLUSH+RELOAD, exploit specific aspects of cache behavior. FLUSH+RELOAD involves flushing a shared cache line and monitoring its reloading by another process, revealing access patterns to

© 2025 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

sensitive data. PRIME+PROBE, in contrast, primes the cache by loading it with the attacker's information and then probes to detect whether another process has evicted some of this data [1] [3]. These techniques, particularly PRIME+PROBE & FLUSH+RELOAD, are the focus of this study, and their detection relies on Hardware Performance Counters (HPCs) like cache hits, misses, and timing variations [6] [7] [8]. While these counters provide essential indicators, they are often inadequate in identifying sophisticated and evolving attack patterns, necessitating advanced detection methods. This is where deep learning models become invaluable as they can analyze large datasets to detect complex patterns and subtle indicators of an attack that traditional methods might miss [9] [10] [11]. Their proficiency in learning and adapting to new attack vectors makes them essential in modern cybersecurity, especially for detecting CSCAs.

This study makes the following contributions:

- Provides a comprehensive comparative analysis of six deep learning models for detecting CSCAs, focusing on both PRIME+PROBE & FLUSH+RELOAD scenarios.
- Proposes the use of a hybrid CNN-LSTM model to improve CSCA detection performance.
- Demonstrates the hybrid model consistently outperforming MLP, CNN, RNN, GRU, and LSTM across various attack types, victims, and performance metrics.

Rest of the paper contains Section II that provides an overview of the literature review. The methodology adopted, including the implemented models and the hybrid approach, is detailed in Section III. Section IV presents and analyzes the results of the experiments. The study concludes with key insights and recommendations for future work in Section V.

II. LITERATURE REVIEW

Chiapetta et al. implemented a machine learning technique for detecting FLUSH+RELOAD CSCA in a cross VM environment using *perf* [7]. Depoix and Altmeyer used machine learning to detect realtime Spectre attacks by collecting performance data using the PAPI tool [12]. Payer proposed a detection system, HexPADS, to detect PRIME+PROBE & FLUSH+RELOAD attacks based on system behavior, specifically distinguishing between normal and abnormal activities using a threshold determined by cache-misses [13]. Briongos et al. proposed a self monitoring tool, CacheShield, to detect FLUSH+FLUSH, PRIME+PROBE &

FLUSH+RELOAD attacks in a cloud-based environment, demonstrating a perfect detection accuracy of 100% [14]. Cho et al. introduced a realtime identification framework for CSCA by monitoring in CPU counters using machine learning algorithms, effectively detecting attacks immediately as they occur [15]. Su and Zeng proposed a security model that evaluates CSCAs based on vulnerability, cache type, pattern, and range, while also exploring defense strategies [16]. Alam et al. developed a machine-learning-based identification system for micro-architectural side-channel attacks (SCAs), utilizing performance counter profiling and time-series analysis and achieved a high detection accuracy of 98.7% [17]. Wang et al. introduced HybriDG, which is a hybrid model combining Gaussian distribution & Dynamic Time Warping to detect both recognized and zero-day microarchitectural SCAs in real-time, achieving a detection accuracy of 99.5% [18]. Mushtaq et al. proposed WHISPER, a machine learning tool for detecting real-time SCAs like FLUSH+FLUSH, FLUSH+RELOAD, Meltdown, PRIME+PROBE & Spectre using HPCs and achieved a remarkable detection accuracy of over 99% [19]. Wang et al. introduced Hybrid-Shield, a cross-layer solution for detecting and mitigating cache-based side-channel attacks, achieving 100% detection accuracy with a Decision Tree Classifier [20].

Le et al. proposed a real-time detection method for Spectre attacks on RISC-V using HPCs and a Neural Network, achieving over 99% accuracy with minimal performance impact [21]. Maheswari and Krishnamurthy brought forth a deep-learning SCA detection method by employing a Deep Residual Capsule Auto-Encoder (DR_CAE) model for attack classification, achieving high performance with an accuracy of 98.80% [22]. Kim et al. introduced a method called FRIME, based on deep-learning for detecting multiple CSCAs, including FLUSH+RELOAD, PRIME+ABORT & PRIME+PROBE, by leveraging both cache as well as Intel TSX-based hardware events. Their LSTM-based model demonstrated superior performance with a detection accuracy of 98.81% for FLUSH+RELOAD and 85.33% for PRIME+PROBE, outperforming their MLP and RNN models [23].

Shang et al. proposed a CNN-LSTM hybrid model for processing ultrasonic guided waves in metallic pipelines, achieving damage detection accuracy of 94.8% as compared to standalone CNN and LSTM models [24]. Han et al. explored a hybrid CNN-LSTM model for time-series data prediction,

TABLE I: RECENT RESEARCH IN CACHE SIDE-CHANNEL ATTACKS' DETECTION

Models	Cache Attack(s)	Victim(s)	Perf. Metrics	
Machine Learning Models [19]	PRIME+PROBE, FLUSH+RELOAD, FLUSH+FLUSH	AES	Accuracy, Speed, FP, FN, Overhead	
MLP [21]	PRIME+PROBE, FLUSH+RELOAD	AES	Accuracy, FP, FN	
Change point detection [14]	PRIME+PROBE, FLUSH+RELOAD, FLUSH+FLUSH	AES, RSA	Detection time, FP Rate	
RNN, MLP, LSTM [23]	PRIME+PROBE, FLUSH+RELOAD, PRIME+ABORT	AES, RSA	Accuracy, FP, FN, Precision, Recall	
OneR, MLP, DT, J48, BayesNet [20]	PRIME+PROBE, FLUSH+RELOAD	AES, RSA	Accuracy, False Alarm Rate	
Machine Learning [12]	FLUSH+RELOAD	Wordpress, Ghost, Chrome	Accuracy, F1-score	

addressing the long-term dependency issue inherent in RNNs [25]. This model showed improved performance in predictive accuracy and power, particularly in dynamic time-series scenarios.

Till now, multiple models have emerged for the detection of CSCAs using machine learning. However, these traditional methods often rely on predefined features and may struggle to adapt to evolving attack patterns. In contrast, deep learning approaches are particularly beneficial because they can autonomously identify and learn intricate patterns from vast datasets without extensive human intervention. Only recently have detection techniques utilizing deep learning models surfaced, addressing the limitations of earlier approaches and offering improved adaptability and accuracy.

Table I summarizes the existing findings, detailing models, cache attacks, targeted victims, and performance metrics.

The contribution of this study is driven by the need to improve detection performance and identify the most effective models for CSCA detection. A comparative analysis of various models is crucial to determine the best-performing approaches. Additionally, this study introduces a new hybrid model approach that offers robust protection against these sophisticated threats, comparing it with existing models to solidify its effectiveness.

III. PROPOSED METHODOLOGY

This section introduces the proposed hybrid model CNN-LSTM and explores five other deep learning architectures, evaluating their effectiveness against CSCAs. A thorough comparative analysis is conducted to explore the trade-offs and benefits of each model, aiming to identify the optimal configurations that enhance security and detection accuracy without compromising system performance.

A. Implemented Models

This subsection reviews the deep-learning architectures used in this study, detailing their structure, primary applications, strengths, and limitations. Each model has been implemented for a thorough comparative analysis in the context of CSCA detection [10].

Convolutional Neural Networks (CNN): CNNs are effective for detecting structured patterns in data. Their ability to capture spatial hierarchies within this data makes them suitable for identifying subtle variations indicative of an attack [8]. While CNNs can capture local patterns, they don't inherently capture sequential dependencies within the series that may be crucial for detecting attacks patterns [26] [27].

Recurrent Neural Networks (RNN): RNNs are designed to efficiently analyze sequential data, making them applicable for monitoring sequences of cache accesses in SCAs. They excel in learning temporal dependencies, which is crucial for identifying attack patterns over time. Nevertheless, traditional RNNs face challenges like vanishing gradients, which can impair their performance in detecting long-term dependencies [8] [11] [28].

Long Short-Term Memory (LSTM): LSTM networks identify long-term dependencies in sequential data, making them well-suited for recognizing patterns in time-series data, like in CSCAs. Their ability to retain information over extended sequences allows them to effectively model the temporal dynamics of loads, misses, and instructions, which are crucial for identifying attack behaviors. Although LSTMs require more computational resources compared to simpler models, their strength in handling sequence data justifies their use in this context [8] [11] [29].

Gated Recurrent Units (GRU): GRUs offer a streamlined architecture compared to LSTM net-

works while retaining the capability to model longterm dependencies in sequential data. Their simplified design enhances computational efficiency, making GRUs a practical choice for detecting CSCAs with reduced resource consumption [9]. Despite their efficiency, GRUs still require significant computational resources and may exhibit performance trade-offs relative to the more complex LSTM architecture [30].

Multi-Layer Perceptrons (MLP): MLPs excel at capturing complex non-linear patterns in static data due to their fully connected layers [9] [31]. They are effective for tasks where feature interactions are crucial, making them relevant for analyzing extracted features in CSCAs. However, MLPs cannot capture temporal dependencies, limiting their performance on sequential tasks compared to LSTMs, GRUs, and RNNs. Their efficiency and simplicity come at the cost of missing dynamic temporal patterns [32].

B. Proposed Hybrid Model

The proposed hybrid architecture integrates CNN and LSTM networks to enhance the detection of CSCAs. This model leverages CNN's capability to extract spatial features from memory access traces, identifying local patterns of loads, misses, and instruction counts. These spatial features are then processed by the LSTM, which capture temporal dependencies and complex patterns over time. By combining the strength of CNN in spatial analysis with LSTM's ability to model temporal dynamics, this approach aims to improve both accuracy and performance in detecting sophisticated attacks. The hybrid model is anticipated to offer superior detection capabilities compared to models using either CNN or LSTM [10] [11] [24] [25].

This model is proposed on observing the diverse set of patterns in the frequency graphs of the data points of the studied attacks. The peaks for the counters of benign and malicious processes are overlapping in some and not in others. On the other hand relatively non-distinct peaks for malicious processes are also observed. These patterns are difficult to detect with a short sequence of inputs, this calls for the use of sequential models. However, distinct peaks between the two are also observed in some cases. Spatial models excel in capturing these instances with their ability to focus on a sliding window of the input. The combination of these abilities can enhance detection.

Fig. 1 illustrates the flow of data through the hybrid CNN-LSTM model through a block diagram. The model processes input sequences by first

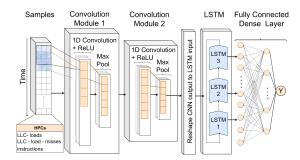


Fig. 1: Block Diagram of Proposed Hybrid Model

passing them through CNN layers, which extract local spatial features using 1D convolution. ReLU activations introduce non-linearity, and max pooling reduces dimensionality while highlighting key features. The CNN output is then fed into LSTM layers, which capture temporal dependencies and long-term patterns within the sequence. Finally, the hidden states from the LSTM are passed through a fully connected layer to produce class predictions based on the combined spatial and temporal features.

This architecture's main strength is its ability to leverage CNNs for spatial feature detection and LSTMs for temporal pattern recognition, making it well-suited for sequence-based tasks.

IV. RESULTS AND DISCUSSION

This section outlines the implementation details, outcomes, and analysis of the study.

A. Experimental Setup

This section explains the environment in which dataset collection, model training and model evaluation is done.

Assumptions & Considerations: Intel i7-11800H processor with a 16GB RAM and 4MB of Last Level Cache (LLC) is utilized for training and evaluating the models. Processor events are collected on Ubuntu 22.04 using Performance Monitoring Units (PMUs). All the experiments and dataset collection are done on Ubuntu 22.04.

Performance Events: Monitoring specific performance events is crucial for detecting anomalies in CSCAs.

The *instructions* metric counts the total number of processor instructions executed within a specific time frame. A significant reduction in instruction count coupled with other relevant readings, may indicate malicious activity [12].

LLC-load-misses occur when data that is not present in the LLC is requested, leading to a cache

miss. Spikes in LLC-load-misses can indicate irregular access patterns that may suggest the presence of a CSCA [13] [33].

LLC-loads refer to read operations accessing data from the LLC. By tracking LLC-loads, one can assess overall cache usage and identify unusual behavior which when studied with LLC-load-misses, can help in detecting anomalies in cache access patterns.

B. Dataset

Victim processes running AES and RSA encryption and decryption, are monitored for live data collection [34] [35]. Malicious implementations, including PRIME+PROBE & FLUSH+RELOAD, are also observed. Monitoring is done using the *perf* tool, which leverages PMUs to gather performance data [36].

perf offers a number of performance events for monitoring purposes. Out of these 2,402 events, three processor events are chosen by considering the runtime behavior of PRIME+PROBE & FLUSH+RELOAD implementations. The selected events are LLC-load-misses, LLC-loads and instructions.

Performance counters for each process are tracked individually, instead of relying on overall CPU data [7]. This distinction enables the model to decide whether a process is harmless or potentially malicious. From there, the detection system responds to each process according to the model's predictions.

Overall, 30,000 datapoints are collected and split into a 50-50 distribution, labeled as benign and malicious, respectively.

Fig. 2 presents results of FLUSH+RELOAD attack during AES and RSA encryption and decryption, respectively [37].

Fig. 3 displays the results of PRIME+PROBE attack under the same conditions.

C. Evaluation Metrics

Accuracy, Precision, Recall, FN and FP are some of the widely used metrics for evaluating detection models for CSCAs [20] [21] [23] [38].

Accuracy and Precision: Accuracy measures correct classifications, ensuring overall model reliability. Precision ensures detected CSCAs are true, reducing false alarms.

Recall: Recall is critical for capturing as many CSCAs as possible, minimizing the risk of undetected threats.

False Positives (FP) and False Negatives (FN): FP indicates when normal processes are incorrectly flagged as attacks, while FN shows when actual

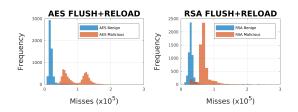


Fig. 2: LLC-load-misses of FLUSH+RELOAD

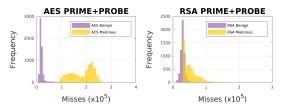


Fig. 3: LLC-load-misses of PRIME+PROBE

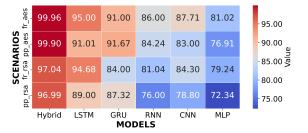


Fig. 4: Heatmap of accuracies of different models

attacks are missed. Both metrics are crucial for evaluating if the model detects all attacks and avoids unnecessary alerts.

D. Results and Comparison

Table II illustrates the results of all the implemented and evaluated models. Each model is tested on its ability to detect PRIME+PROBE & FLUSH+RELOAD attacks on both AES and RSA as victim programs. The table has four sections to illustrate the results of each type of test viz. FLUSH+RELOAD (AES), FLUSH+RELOAD (RSA), PRIME+PROBE (AES), and PRIME+PROBE (RSA). Each model is evaluated on five metrics which are Accuracy, Precision, Recall, FP & FN.

Fig. 4 shows the heatmap of model accuracies across various scenarios and models.

Two clear patterns are observed within model performance throughout attacks and environments. Firstly, the general pattern observed is that attacks on RSA have a lower detection rate across the board, with accuracy dropping anywhere between 2.00% to 8.00% in models detecting the same type of attack. This could be attributed to the fact that

TABLE II: COMPARISON OF DIFFERENT MODELS UNDER DIVERSE ATTACK SCENARIOS

	Model	Accuracy	Precision	Recall	FP	FN
FLUSH + RELOAD AES	MLP	81.02	89.51	89.51	9.49	9.49
	CNN	87.71	92.78	94.06	6.83	5.46
	RNN	86.00	86.58	99.23	13.33	0.67
	GRU	91.00	98.91	91.92	1.00	8.00
	LSTM	95.00	99.99	95.01	0.01	4.99
	Hybrid (CNN-LSTM)	99.96	99.99	99.96	0.00	0.04
FLUSH + RELOAD RSA	MLP	79.24	88.42	88.42	10.38	10.38
	CNN	84.30	90.13	92.88	9.24	6.46
	RNN	81.04	81.64	99.11	18.23	0.73
	GRU	84.00	97.90	85.54	1.80	14.20
	LSTM	91.01	94.51	96.09	5.29	3.70
	Hybrid (CNN-LSTM)	99.90	99.95	99.95	0.05	0.05
PRIME + PROBE AES	MLP	76.91	87.33	86.57	11.15	11.94
	CNN	83.00	93.61	87.99	5.67	11.33
	RNN	84.24	89.04	93.98	10.37	5.39
	GRU	91.67	99.99	91.67	0.00	8.33
	LSTM	94.68	99.99	94.69	0.01	5.31
	Hybrid (CNN-LSTM)	97.04	99.99	97.05	0.01	2.95
PRIME + PROBE RSA	MLP	72.34	83.95	83.95	13.83	13.83
	CNN	78.80	91.51	85.01	7.31	13.89
	RNN	76.00	81.91	91.33	16.78	7.22
	GRU	87.32	99.99	87.32	0.00	12.68
	LSTM	89.00	93.37	95.01	6.32	4.68
	Hybrid (CNN-LSTM)	96.99	98.61	98.34	1.37	1.64

RSA has a more irregular access pattern than an AES system [34]. This results in higher variability in cache miss patterns, making it difficult to distinguish benign activity from malicious activity, i. e. FP in RSA scenarios are much higher [37]. The same is reflected in the table with gaps up to 6.00% in the FP values. The attack patterns are muddled by the inconsistent cache behaviour, making it increasingly difficult for spatial models like MLP, CNN; and to a certain extent short sequence temporal models like RNN and GRU to extract meaningful insights from the HPCs collected. The attack pattern distortion created in such RSA environments can only be combated by studying these irregularities on a more longterm scale, hence allowing for effective attack pattern isolation.

Secondly, this study reveals distinct patterns in the performance of each model, with MLP consistently achieving the lowest accuracy across all test types. However, it exhibits a relatively standout performance in FLUSH+RELOAD scenarios, reaching 81.02% accuracy, alongside a precision of 89.51%. There is a clear divide between the accuracy rates it is able to achieve in AES and RSA scenarios, with a gap of roughly 4.00% in FLUSH+RELOAD and 7.00% in PRIME+PROBE; one of the highest disparities amongst all models.

CNN performs slightly better with 87.71% accuracy for FLUSH+RELOAD (AES) but struggles with PRIME+PROBE, especially in the RSA scenario, where it records 78.80% accuracy. This indicates that despite strong feature recognition, it has limitations in capturing intricate temporal patterns [27]. It is observed that PRIME+PROBE cache miss patterns occur in bursts or clusters as the attacker probes multiple cache sets, spreading misses periodically over time. In contrast, FLUSH+RELOAD produces localized misses tied to specific cache lines, forming a targeted spatial pattern rather than a temporal pattern.

Hence, spatial models like MLP and CNN, struggle to detect PRIME+PROBE but fare better in detecting FLUSH+RELOAD. CNN overtakes MLP in the former due to its minimal pattern recognition ability being applied when time is taken as one of the spatial dimensions.

RNN performs slightly worse than CNN, with accuracy dropping to 86.00% for FLUSH+RELOAD (AES) due to its inconsistent spikes. However, it reduces the FN rates by 6%, exhibiting improved detection of attack patterns. The FP rates increase by roughly 9%, indicating difficulty in distinguishing attack patterns from benign activity. The vanishing gradient issue likely contributes to these results,

as RSA patterns require long-term memory and complex pattern recognition in order to distinguish it from malicious activity.

GRU offers a more balanced performance, achieving roughly 91.00% accuracy for both PRIME+PROBE & FLUSH+RELOAD in AES, with up to 98.91% precision and a roughly equivalent recall of 91.92% It significantly improves both FN and FP rates, indicating a strong ability to differentiate between benign and malicious processes. Notably, it outperforms other models in PRIME+PROBE tests, achieving 87.32% accuracy for PRIME+PROBE (RSA), where others stagnate at roughly 72.00%, signalling effective modeling of PRIME+PROBE's temporal attack patterns.

The LSTM model records a 94.68% accuracy for PRIME+PROBE (AES) and 95.00% for FLUSH+RELOAD (AES), hinting at its improved performance in temporal pattern recognition over spatial pattern recognition. The FN and FP rates for LSTM remain notabily balanced, showcasing its strength in retaining the afformentioned long-term dependencies in the patterns of PRIME+PROBE and RSA. Unlike GRU, it does not struggle with pattern distinguishing tasks and performs uniformly if not better.

The proposed hybrid CNN-LSTM model outperforms all other models in both PRIME+PROBE & FLUSH+RELOAD attacks, across both RSA and AES scenarios. It reaches an exceptional 99.96% accuracy for FLUSH+RELOAD (AES) and 99.00% for FLUSH+RELOAD (RSA), with the lowest yet FP and FN rates ranging from 0.05% in AES to a maximum of 3.00% in RSA. The precision value peaks at 99.99%, validating the hybrid model's superior ability to capture both spatial and temporal patterns in detecting CSCAs. It hence demonstrates not only pattern recognition but also pattern differentiation, which the previous models do not have the capability to do due to their highly specialized applications allowing only one of these functions at a time.

This clearly outlines the robustness of the hybrid approach and the value of combining the functional competencies of CNN and LSTM architectures for optimal performance.

The proposed hybrid CNN-LSTM model outperforms the LSTM model implementation by Kim et al. in detecting both PRIME+PROBE & FLUSH+RELOAD attacks [23]. This work is preferred for its extensive coverage of attack scenarios, achieving 98.81% accuracy for FLUSH+RELOAD (RSA) and 85.33% for PRIME+PROBE (AES). In contrast, the proposed hybrid model achieves

99.01% and 97.04% accuracy, respectively. The model also records 99.96% for FLUSH+RELOAD (AES) and 96.99% accuracy for PRIME+PROBE (RSA), demonstrating superior performance across multiple CSCAs.

V. CONCLUSION AND FUTURE SCOPE

This study demonstrates the effectiveness of the hybrid CNN-LSTM model in detecting various CSCA scenarios. It combines the spatial feature extraction abilities of CNN with the LSTM's strength in modeling temporal dynamics. The hybrid approach achieves superior performance in detecting both PRIME+PROBE & FLUSH+RELOAD attacks, with higher accuracy and precision than all five standalone models. It achieved an overall accuracy of 99.96%, significantly outperforming the implemented standalone models while also maintaining consistent performance across diverse attack types and environments. This marks a substantial advancement in CSCA detection. Moreover, this study, in comparison to the extensive literature studied in this domain, is amongst the first to provide a robust comparative analysis of these deep-learning architectures in CSCA detection. It is also the first to introduce the use of a proficient hybrid model for detection. It condenses decades of research to extract meaningful insights that could propel further development. Another key takeaway from this study is that the long-term pattern recognition capabilities of temporal models like LSTM, GRU, and RNN are enhanced when combined with the localized pattern recognition strengths offered by spatial models such as CNN and MLP.

Further study could focus on expanding the model's detection abilities to take into account a more diverse set of HPCs. This may allow for the detection of a broader range of attacks. Moreover, future research can focus on optimizing the hybrid architecture to enhance computational efficiency without compromising its performance. Achieving this balance will be crucial in ensuring that the model remains both effective and practical for real-world application scenarios.

REFERENCES

- Yarom, Yuval, and Katrina Falkner. "FLUSH+ RELOAD: A high resolution, low noise, 13 cache Side-Channel attack." 23rd USENIX security symposium (USENIX security 14). 2014.
- [2] Lyu, Yangdi, and Prabhat Mishra. "A survey of sidechannel attacks on caches and countermeasures." Journal of Hardware and Systems Security 2 (2018): 33-50.
- [3] Liu, Fangfei, et al. "Last-level cache side-channel attacks are practical." 2015 IEEE symposium on security and privacy. IEEE, 2015.

- [4] Zhang, Yinqian, et al. "Cross-tenant side-channel attacks in PaaS clouds." Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014
- [5] Su, Chao, and Qingkai Zeng. "Survey of CPU Cache-Based Side-Channel Attacks: Systematic Analysis, Security Models, and Countermeasures." Security and Communication Networks 2021.1 (2021): 5559552.
- [6] Khoshavi, Navid, et al. "A survey on attack vectors in stack cache memory." Integration 72 (2020): 134-147.
- [7] Chiappetta, Marco, Erkay Savas, and Cemal Yilmaz. "Real time detection of cache-based side-channel attacks using hardware performance counters." Applied Soft Computing 49 (2016): 1162-1174.
- [8] Kulah, Yusuf, et al. "SpyDetector: An approach for detecting side-channel attacks at runtime." International Journal of Information Security 18 (2019): 393-422.
- [9] Shiri, Farhad Mortezapour, et al. "Deep learning in big data analytics: A comparative study." ACM Computing Surveys 56.2 (2023): 1-40.
- [10] Shiri, Farhad Mortezapour, et al. "A comprehensive overview and comparative analysis on deep learning models: CNN, RNN, LSTM, GRU." arXiv preprint arXiv:2305.17473 (2023).
- [11] Shiri, Farhad Mortezapour, et al. "Recent advances in deep learning models: A systematic literature review." arXiv preprint arXiv:2305.17473 (2023).
- [12] Depoix, Jonas, and Philipp Altmeyer. "Detecting spectre attacks by identifying cache side-channel attacks using machine learning." Advanced Microkernel Operating Systems 75 (2018): 48.
- [13] Payer, Mathias. "HexPADS: a platform to detect "stealth" attacks." Engineering Secure Software and Systems: 8th International Symposium, ESSoS 2016, London, UK, April 6–8, 2016. Proceedings 8. Springer International Publishing 2016.
- [14] Briongos, Samira, et al. "Cacheshield: Detecting cache attacks through self-observation." Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy. 2018.
- [15] Cho, Jonghyeon, et al. "Real-time detection for cache side channel attack using performance counter monitor." Applied Sciences 10.3 (2020): 984.
- [16] Su, Chao, and Qingkai Zeng. "Survey of CPU Cache-Based Side-Channel Attacks: Systematic Analysis, Security Models, and Countermeasures." Security and Communication Networks 2021.1 (2021): 5559552.
- [17] Alam, Manaar, Sarani Bhattacharya, and Debdeep Mukhopadhyay. "Victims Can Be Saviors: A Machine Learning-based Detection for Micro-Architectural Side-Channel Attacks." ACM Journal on Emerging Technologies in Computing Systems (JETC) 17.2 (2021): 1-31.
- [18] Wang, Han, et al. "HybriDG: Hybrid dynamic time warping and Gaussian distribution model for detecting emerging zero-day microarchitectural side-channel attacks." 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). IEEE, 2020.
- [19] Mushtaq, Maria, et al. "WHISPER: A tool for run-time detection of side-channel attacks." IEEE Access 8 (2020): 83871-83900
- [20] Wang, Han, et al. "Hybrid-shield: Accurate and efficient cross-layer countermeasure for run-time detection and mitigation of cache-based side-channel attacks." Proceedings of the 39th International Conference on Computer-Aided Design. 2020.
- [21] Le, Anh-Tien, et al. "A real-time cache side-channel attack detection system on RISC-V out-of-order processor." IEEE Access 9 (2021): 164597-164612.
- [22] Maheswari, Raja, and Marudhamuthu Krishnamurthy. "A profiled side-channel attack detection using deep learning model with capsule auto-encoder network." Transactions

- on Emerging Telecommunications Technologies, vol. 35, no. 4, 2024, p. e4975, https://doi.org/10.1002/ett.4975.
- [23] Kim, Hodong, et al. "Deep Learning-Based Detection for Multiple Cache Side-Channel Attacks." IEEE Transactions on Information Forensics and Security, vol. 19, 2024, pp. 1672–1686, https://doi.org/10.1109/TIFS.2023.3340088.
- [24] Shang, Li, et al. "CNN-LSTM Hybrid Model to Promote Signal Processing of Ultrasonic Guided Lamb Waves for Damage Detection in Metallic Pipelines." Sensors, 9 Aug. 2023, https://doi.org/10.3390/s23167059.
- [25] Han, Chungku, et al. "Hybrid CNN-LSTM based timeseries data prediction model study." IEEE/ACIS International Conference on Big Data, Cloud Computing, and Data Science Engineering. Cham: Springer International Publishing, 2022.
- [26] Li, Zewen, et al. "A survey of convolutional neural networks: analysis, applications, and prospects." IEEE transactions on neural networks and learning systems 33.12 (2021): 6999-7019.
- [27] Zhao, Dapeng, and Jean Oh. "Noticing motion patterns: A temporal cnn with a novel convolution operator for human trajectory prediction." IEEE Robotics and Automation Letters 6.2 (2020): 628-634.
- [28] Medsker, Larry R., and Lakhmi Jain. "Recurrent neural networks." Design and Applications 5.64-67 (2001): 2.
- [29] Sherstinsky, Alex. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." Physica D: Nonlinear Phenomena 404 (2020): 132306.
- [30] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).
- [31] Popescu, Marius-Constantin, et al. "Multilayer perceptron and neural networks." WSEAS Transactions on Circuits and Systems 8.7 (2009): 579-588.
- [32] Mohammadi Foumani, Navid, et al. "Deep learning for time series classification and extrinsic regression: A current survey." ACM Computing Surveys 56.9 (2024): 1-45.
- [33] Junior, Juscelino Candido De Lima, et al. "Identifying malicious executing code of an enclave." U.S. Patent No. 10,997,289. 4 May 2021.
- [34] Osvik, Dag Arne, Adi Shamir, and Eran Tromer. "Cache attacks and countermeasures: the case of AES." Topics in Cryptology-CT-RSA 2006: The Cryptographers' Track at the RSA Conference 2006, San Jose, CA, USA, February 13-17, 2005. Proceedings. Springer Berlin Heidelberg, 2006.
- [35] Aciiçmez, Onur, and Werner Schindler. "A vulnerability in RSA implementations due to instruction cache analysis and its demonstration on OpenSSL." Topics in Cryptology–CT-RSA 2008: The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings. Springer Berlin Heidelberg, 2008.
- [36] De Melo, Arnaldo Carvalho. "The new linux'perf'tools." Slides from Linux Kongress. Vol. 18. 2010.
- [37] Fatima, S., et al. "Comparative Analysis of Aes and Rsa Algorithms for Data Security in Cloud Computing. Eng. Proc. 2022, 20, 14." (2022).
- [38] Powers, David MW. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." arXiv preprint arXiv:2010.16061 (2020).