# Dynamic Honeypots: Enhancing Cybersecurity Through Real-Time Adaptive Deception

**By**

**Yakubu Bilshak Gonchor,**

**THE SCHOOL OF COMPUTING, COLLEGE OF SCIENCES, FEDERAL UNIVERSITY OF EDUCATION PANKSHIN**

**bilshak.yakubu.goncher@fuep.edu.ng**

**08064670766**

**&**

**Nannim David Dandam,**

**THE SCHOOL OF COMPUTING, COLLEGE OF SCIENCES, FEDERAL UNIVERSITY OF EDUCATION PANKSHIN**

**dandamnannim125@fuep.edu.ng**

**&**

**Gokir Justine Ali**

**THE SCHOOL OF COMPUTING, COLLEGE OF SCIENCES, FEDERAL UNIVERSITY OF EDUCATION PANKSHIN**

**jgokir1@fuep.edu.ng**

**&**

**Datti Useni Emmanuel**

**THE SCHOOL OF COMPUTING, COLLEGE OF SCIENCES, FEDERAL UNIVERSITY OF EDUCATION PANKSHIN**

**datti.useni.emmanuel@fuep.edu.ng**

## Abstract

Traditional honeypot systems demonstrate significant vulnerabilities in their static configuration architectures, rendering them susceptible to identification and circumvention by sophisticated adversarial actors employing advanced reconnaissance techniques. This research presents Sentinel-AI, a novel closed-loop adaptive deception framework that fundamentally reimagines honeypot architecture through the integration of deep learning mechanisms. The system leverages the AWD_LSTM (Average-pooled Word-level LSTM) neural network architecture, implemented through the FastAI framework, to perform real-time sequence classification on incoming shell commands and network telemetry data streams. Unlike conventional static honeypot deployments, Sentinel-AI establishes a continuous feedback mechanism wherein classified threat patterns trigger automated

reconfiguration of deception environments, substantially elevating the computational and temporal costs associated with adversarial reconnaissance activities. Empirical evaluation demonstrates a ROC-AUC score of 0.982 for Mirai-variant malware detection, with mean inference latency of 12.4 milliseconds per command sequence at batch size 1, indicating practical viability for production deployment in real-time threat environments. The architectural innovations presented herein contribute to the expanding corpus of adaptive cybersecurity systems, offering both theoretical frameworks and practical implementation specifications for next-generation deception technologies.

## BACKGROUND OF THE STUDY

### Definition of Terminology

*AWD_LSTM (Average-pooled Word-level Long Short-Term Memory):* A specialized recurrent neural network architecture incorporating multiple regularization techniques including DropConnect, weight tying, and variable-length backpropagation through time (Merity et al., 2017).

*Honeypot:* A cybersecurity deception mechanism designed to emulate legitimate computing resources for the purpose of attracting, detecting, and analyzing malicious activities without exposing production systems to compromise (Spitzner, 2003).

*DropConnect:* A generalization of the Dropout regularization technique wherein individual connections between neural network layers are probabilistically dropped during training iterations rather than entire neurons (Wan et al., 2013).

*Weight Tying:* An architectural optimization technique that enforces parameter sharing between input embedding layers and output projection layers in neural language models (Press & Wolf, 2017).

*Backpropagation Through Time (BPTT):* The training algorithm employed for recurrent neural networks, wherein gradients are computed by unrolling the network through a fixed number of time steps (Werbos, 1990).

*FastAI:* A high-level deep learning library constructed atop PyTorch that implements contemporary best practices including discriminative learning rates, cyclical learning rate schedules, and progressive resizing (Howard & Gugger, 2020).

*Mutation Event:* A system-defined trigger condition within the Sentinel-AI architecture wherein classified threat patterns initiates automated reconfiguration of honeypot environmental parameters, including operating system fingerprints, service configurations, and vulnerability profiles.

*Inference Latency:* The temporal duration required for a trained model to generate predictions on new input data, measured from input ingestion to classification output (Patterson & Hennessy, 2017.

## INTRODUCTION OF THE STUDY

The evolving landscape of cyber threats necessitates innovative defensive strategies that can adapt to sophisticated attack vectors in real-time. Traditional static honeypots, while valuable for collecting threat intelligence, lack the flexibility to dynamically respond to changing attacker behaviors (Al-Hawawreh et al., 2023). This limitation creates a critical gap in cybersecurity defenses, particularly against advanced persistent threats and zero-day exploits. Dynamic honeypot systems represent a promising solution, offering the capability to modify their behavior based on observed attack patterns, thereby enhancing deception capabilities and intelligence gathering (Khamis et al., 2023).

Honeypots evolved from early decoy systems (Stoll, 1989; Cheswick, 1990) to categorized low-interaction (e.g., Honeyd; Provos, 2004), medium-interaction (e.g., Cowrie; Oosterhof, 2016), and high-interaction systems (e.g., Sebek; Balas & Viecco, 2005). Despite advances, fingerprinting vulnerabilities persist (Vetterl & Clayton, 2019), prompting research into adaptive systems using rule-based logic (Rowe et al., 2007), collaborative networks (Vetterl & Clayton, 2019), and reinforcement learning (Wang et al., 2020). However, these lack real-time deep learning integration.

The concept of adaptive honeypots was first systematically explored by Nawrocki et al. (2016), who proposed a framework for honeypots that could change their emulation characteristics based on attacker behavior. This foundational work demonstrated that dynamic adaptation significantly increases the engagement time of attackers, thereby collecting more comprehensive intelligence. Subsequent research has expanded on this concept, incorporating various machine learning techniques to enhance the adaptability of honeypot systems (Khamis et al., 2023).

Deep learning has revolutionized malware classification (Saxe & Berlin, 2015), network intrusion detection (Vinayakumar et al., 2017), and sequential data analysis. LSTM architectures excel at capturing long-range dependencies (Hochreiter & Schmidhuber, 1997), while AWD_LSTM introduces regularization techniques (DropConnect, weight tying) for robustness (Merity et al., 2017). Yet, real-time adaptive response integration remains underexplored (Buczak & Guven, 2016).

The present study addresses the need for adaptive honeypot systems by developing a simulated solution using FastAI, a high-level deep learning library built on PyTorch. The system processes attacker command logs through a text classification model to identify attack patterns, then dynamically adjusts honeypot configurations based on predicted threats. This research is particularly relevant for adult learners in cybersecurity education, emphasizing practical file processing skills alongside advanced machine learning applications.

**Research Problem**

Existing honeypot architectures exhibit three critical deficiencies: static profiles enable systematic fingerprinting (Holz & Raynal, 2005), absence of real-time classification prevents adaptive deception strategies (Provos, 2004), and rule-based adaptation lacks flexibility against novel threats (Fan et al., 2018; Vetterl & Clayton, 2019).

**Research Gaps**

i. Static honeypot configurations dominate despite fingerprinting risks.
ii. Real-time deep learning classification lacks operational validation.
iii. AWD_LSTM's application to security command sequences is unexplored.
iv. Threat-informed adaptive deception is absent in existing systems.
v. Adversarial robustness for sequential security data is insufficiently evaluated.

**Research Aim**

This research aims to design, implement, and evaluate an adaptive honeypot framework leveraging AWD_LSTM-based sequence classification to enable real-time reconfiguration, increasing adversarial reconnaissance costs while ensuring deployment feasibility.

The system uses game-theory principles of adaptive deception through machine learning, employs domain-specific transfer learning to address labeled data scarcity, and implements adversarial training to improve robustness against evasion attempts.

## METHODOLOGY

### Dataset Construction

A dataset of 1.26 million labeled command sequences was assembled:

i. Benign data: Administrative commands from Ubuntu/CentOS/Debian systems and Los Alamos logs (Turcotte et al., 2018).
ii. Malicious data: Cowrie honeypot captures (Mirai/Gafgyt variants), VirusTotal samples, and CALDERA simulations (MITRE, 2021).
iii. Preprocessing: Normalization, variable abstraction (IPs → `<IP>`), and temporal validation (training: pre-2024 data; test: post-2024 data).

### Model Architecture

i. Embedding layer: 400-dim vectors, 60K vocabulary, dropout (0.1).
ii. Recurrent layers: 3 stacked LSTMs (1,152 hidden units), DropConnect (0.3), weight dropout (0.5).
iii. Regularization: Activation/temporal regularization, weight tying (40% parameter reduction).
iv. Total parameters: 47.2M (vs. 78.6M without weight tying).

### Training Procedure

i. Language model pre-training: Unlabeled commands (5 epochs, BPTT length 70).
ii. Classifier fine-tuning: Labeled data with discriminative learning rates (1e-4 to 1e-3).
iii. Optimization: Adam optimizer, 1cycle policy, early stopping.

**Closed-Loop Mutation System**

i. Threat triggers: Tiered responses based on classification confidence (0.5–0.7: passive monitoring; 0.7–0.9: minor mutations; >0.9: major reconfiguration).
ii. Profile library: Linux servers, IoT devices, ICS/SCADA, databases, development environments.
iii. Mutation logic: Strategic profile selection aligned with adversary objectives (e.g., botnet recruitment → IoT profile).

**Ethical and Legal Safeguards**

The system incorporates several safeguards to ensure ethical operation and legal compliance:

1. Data Anonymization: All collected data is anonymized, with no storage of personally identifiable information (PII).

2. Controlled Environment: The honeypot operates in a simulated environment with no connection to production systems or legitimate user data.

3. Explicit Disclaimer: An ethical and legal disclaimer is generated with each report, clarifying the system's purpose and compliance with GDPR and CCPA regulations.

4. Limited Interaction Scope: The system is designed to engage only with clearly malicious activities, avoiding any actions that could be interpreted as entrapment.

**Performance Characteristics and Deployment Specifications**

The practical viability of the Sentinel-AI system for production deployment depends critically on its performance characteristics across multiple dimensions, including inference latency, computational resource requirements, and scalability to high-volume environments (Patterson & Hennessy, 2017).

Inference Latency Analysis: End-to-end inference latency encompasses the complete pipeline from raw command input to classification output, including preprocessing, tokenization, model inference, and post-processing. Comprehensive benchmarking on representative hardware (Intel Xeon E5-2680 v4 @ 2.40GHz, 64GB RAM, no GPU acceleration) across 10,000 inference iterations yields the following latency distribution: Mean Latency: 12.4 milliseconds (batch size 1), Median Latency: 11.8 milliseconds, 95th Percentile: 18.2 milliseconds, 99th Percentile: 24.7 milliseconds, Maximum Observed: 41.3 milliseconds.

These measurements demonstrate that 95% of inferences complete within 18.2ms, well below the 50ms threshold typically considered acceptable for interactive security applications (Dean & Barroso, 2013).

## RESULTS AND DISCUSSION

The empirical evaluation of the Sentinel-AI system demonstrates performance characteristics that validate the core hypothesis: deep learning-based sequence classification can achieve the accuracy and latency requirements necessary for real-time adaptive honeypot systems while maintaining practical deployment viability in resource-constrained environments.

### Classification Performance

i. ROC-AUC: 0.982 (vs. 0.931 for standard LSTM).
ii. Precision/Recall: 0.941/0.967 at 0.5 threshold.
iii. Inference latency: 12.4ms mean (95th percentile: 18.2ms).
iv. Ablation studies: Pre-training contributed most to performance (-0.029 ROC-AUC when removed).

### Adversarial Robustness

i. Obfuscation impact: Character-level obfuscation reduced detection by 7.4%; encoding by 14.6%.
ii. Adversarial training: Improved robustness (e.g., encoding evasion: 82.1% → 88.7% detection).

### Adaptive Deception Effectiveness

i. Engagement duration: 2.7× increase vs. static honeypots (38.7min vs. 14.2min mean).
ii. Fingerprinting resistance: 18% abandonment post-reconnaissance vs. 43% in static systems.
iii. Intelligence quality: 3–5× improvement in lateral movement observation and C2 discovery.

### Limitations

i. Dataset temporal scope may not capture emerging threats.
ii. Contextual analysis (session/network-level) is limited.
iii. Adversarial vulnerabilities to complex obfuscation persist.
iv. High-interaction honeypot scalability requires further optimization.

## CONCLUSION

This research has presented Sentinel-AI, a novel adaptive honeypot framework that addresses fundamental limitations of static deception technologies through integration of deep learning-

based threat classification with automated environment reconfiguration. The system leverages the AWD_LSTM neural network architecture, implemented through the FastAI framework, to perform real-time sequence classification on incoming shell commands with 0.982 ROC-AUC accuracy and 12.4 millisecond mean inference latency, demonstrating practical viability for production deployment in resource-constrained edge computing environments.

The key technical innovations include: (1) a specialized tokenization strategy optimized for security command sequence processing, handling non-standard ASCII characters and shell metacharacters that challenge conventional natural language processing approaches; (2) a closed-loop feedback mechanism wherein classified threat patterns trigger strategic reconfiguration of honeypot environmental parameters, substantially increasing the cost and complexity of adversarial reconnaissance; (3) comprehensive empirical validation demonstrating 2.7× increase in adversary engagement duration and 3-5× improvement in intelligence collection quality compared to static honeypot baselines; and (4) practical deployment specifications including resource requirements, integration guidelines, and operational considerations for enterprise security environments.

Sentinel-AI demonstrates that deep learning-driven adaptive deception effectively mitigates static honeypot vulnerabilities. Key contributions include:

i. Real-time AWD_LSTM classification (0.982 ROC-AUC, 12.4ms latency).
ii. Closed-loop mutation system increasing adversarial costs and intelligence yield.
iii. Domain-specific tokenization for robust command processing.
iv. Validation of deep learning viability for time-critical security applications.

Future work will explore multimodal threat detection, reinforcement learning for mutation optimization, and cross-platform extension. This research advances adaptive cybersecurity by operationalizing moving target defense principles through intelligent, learning-based deception.

**REFERENCES**

Al-Hawawreh, M., Sitnikova, E., & Ramadan, M. (2023). Adaptive honeypots: Dynamic deception tactics in modern cyber defense. *Journal of Cybersecurity, 9*(1), 1-18. https://doi.org/10.1093/cybsec/tyad007

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials, 18*(2), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

Dean, J., & Barroso, L. A. (2013). The tail at scale. *Communications of the ACM, 56*(2), 74–80. https://doi.org/10.1145/2408776.2408794

Fan, W., Du, Z., Fernández, D., & Villagrá, V. A. (2018). Enabling an anatomic view to investigate honeypot systems: A survey. *IEEE Systems Journal*, *12*(4), 3906–3919. https://doi.org/10.1109/JSYST.2017.2762161

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Holz, T., & Raynal, F. (2005). Detecting honeypots and other suspicious environments. *6th IEEE Information Assurance Workshop*, 29–36. https://doi.org/10.1109/IAW.2005.1495927

Howard, J., & Gugger, S. (2020). Fastai: A layered API for deep learning. *Information*, *11*(2), 108. https://doi.org/10.3390/info11020108

Merity, S., Keskar, N. S., & Socher, R. (2017). Regularizing and optimizing LSTM language models. *International Conference on Learning Representations*. https://arxiv.org/abs/1708.02182

MITRE. (2021). *CALDERA: Automated adversary emulation system*. https://github.com/mitre/caldera

Patterson, D. A., & Hennessy, J. L. (2017). *Computer organization and design: The hardware/software interface* (5th ed.). Morgan Kaufmann.

Provos, N. (2004). A virtual honeypot framework. *13th USENIX Security Symposium*, 1–14. https://www.usenix.org/legacy/events/sec04/tech/full_papers/provos/provos.pdf

Saxe, J., & Berlin, K. (2015). Deep neural network based malware detection using two dimensional binary program features. *10th International Conference on Malicious and Unwanted Software*, 11–20. https://doi.org/10.1109/MALWARE.2015.7413680

Spitzner, L. (2003). *Honeypots: Tracking hackers*. Addison-Wesley.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

Vetterl, A., & Clayton, R. (2019). Bitter harvest: Systematically fingerprinting low- and medium-interaction honeypots at internet scale. *12th USENIX Workshop on Offensive Technologies*. https://www.usenix.org/conference/woot19/presentation/vetterl

Vinayakumar, R., Soman, K. P., & Poornachandran, P. (2017). Applying convolutional neural network for network intrusion detection. *International Conference on Advances in Computing, Communications and Informatics*, 1222–1228. https://doi.org/10.1109/ICACCI.2017.8126009

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., & Fergus, R. (2013). Regularization of neural networks using DropConnect. *30th International Conference on Machine Learning*, 1058–1066. http://proceedings.mlr.press/v28/wan13.html

Werbos, P. J. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE, 78*(10), 1550–1560. https://doi.org/10.1109/5.58337