www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

AI-Driven Network Monitoring: A Smart Approach Using SNMP and ML Integration

Luca Bianchi

Digital Infrastructure & Security Center, University of Bologna, Italy

Abstract
Traditional network monitoring systems rely on static thresholds and reactive alerting based on SNMP polling or trap notifications. This paper introduces a smart network monitoring architecture that integrates SNMP data with machine learning models to detect anomalies and forecast performance degradation proactively. Using a dataset of over 2 million SNMP records collected from routers, switches, and firewalls across a university campus network, we extract features such as CPU load, interface utilization, error rates, and packet drop trends. Long Short-Term Memory (LSTM) networks are trained per device type to model baseline behavior and detect deviations in time-series metrics. The system achieves 95.4% precision and 92.1% recall in detecting congestion buildup, hardware failures, and abnormal traffic patterns. Compared to static threshold alerts, the ML-based model reduces false positives by 48% and enables early detection of 70% of incidents before threshold crossing. A dashboard interface built on Grafana displays predicted versus actual metrics, anomaly scores, and root cause indicators. Deployment in a pilot setting demonstrates that the model adapts to dynamic baselines and provides actionable insights with minimal retraining. We also discuss integrating the system with NetFlow and syslog data to enhance correlation. The study concludes with a blueprint for transitioning from legacy SNMP-only monitoring to intelligent, ML-enhanced observability.

---

1. Introduction

Network monitoring has long relied on Simple Network Management Protocol (SNMP) polling and trap-based alerting systems to identify performance bottlenecks and operational anomalies. While effective for basic fault detection, these systems often operate on rigid static thresholds, which struggle to adapt to the dynamic nature of modern network environments. As a result, administrators are either overwhelmed by false positives or left unaware of issues that develop gradually and cross thresholds too late.

This paper explores the integration of SNMP-based telemetry with machine learning (ML) models—specifically Long Short-Term Memory (LSTM) neural networks—to transition from static alerting to predictive, context-aware monitoring. We propose a hybrid architecture that preserves SNMP compatibility while enabling time-series modeling to detect performance degradation before it manifests as service-impacting failures.

Our implementation is validated in a university campus network with thousands of monitored devices, enabling us to test both accuracy and generalizability across diverse network roles. By leveraging ML models trained on historical SNMP metrics, the system continuously evaluates real-time data for anomalies and offers predictive insights, dramatically improving incident response times and reducing alert noise.

2. Related Work

Traditional SNMP monitoring platforms such as Nagios, Zabbix, and SolarWinds use fixed thresholds configured per metric or device, triggering alerts when values exceed predefined limits. While easy to implement, this method is reactive and often fails to capture gradual degradation or distinguish between benign spikes and true anomalies. Moreover, misconfigured thresholds can produce excessive noise, reducing the effectiveness of operational teams.

Recent research has explored applying ML to time-series data for anomaly detection in various domains, including cloud infrastructure and application monitoring. Studies by Chen et al. (2020) and He et al. (2016) introduced LSTM-based models for system behavior modeling, showing strong performance in capturing sequential dependencies and seasonal trends. However, few have integrated ML directly with SNMP-based telemetry due to the coarser temporal resolution and varied polling intervals.

Additionally, open-source ML-integrated monitoring platforms like Prometheus with KubePrometheusStack have begun including rudimentary anomaly detection through exponential smoothing or Holt-Winters forecasting. Yet, these approaches often fall short of capturing non-linear patterns present in real network metrics. Our work builds upon these efforts by applying deep recurrent neural networks and evaluating their practicality in legacy SNMP environments without requiring overhaul of existing infrastructure.

3. Methodology

3.1 Data Collection

- SNMP polling was performed across routers, switches, and firewalls spanning 17 buildings on a university campus.
- 2 million SNMP samples were collected over 45 days using polling intervals of 60 seconds.
- Metrics collected: CPU load, memory usage, interface utilization, input/output errors, discard counts, and packet drop trends.

3.2 Preprocessing

- Time synchronization was enforced across pollers.
- Missing values were interpolated using forward fill and spline techniques.
- Metrics were normalized by device type to account for hardware variability.

3.3 Feature Engineering

- Rolling averages, standard deviations, and rate-of-change features were extracted.
- Time-windowed lag features were introduced for LSTM sequences (lookback = 15 minutes).

3.4 ML Model

- Per-device-type LSTM models trained using Keras with TensorFlow backend.
- Early stopping and dropout applied to avoid overfitting.
- Anomaly scores calculated as deviation between predicted and observed values, normalized via Z-scores.

3.5 Evaluation

- Ground truth labels derived from incident tickets and admin logs.
- Metrics used: precision, recall, F1-score, and alert lead time (time between anomaly detection and threshold breach).

---

4. Experimental Setup and Evaluation Criteria

4.1 Testbed Infrastructure

- Network includes 58 switches, 14 routers, and 6 firewalls across multiple VLANs and zones.
- Monitoring system runs on 2 Ubuntu VMs (8 vCPUs, 16GB RAM) with InfluxDB for storage and Grafana for visualization.
- LSTM inference offloaded to NVIDIA T4 GPU to accelerate batch processing.

4.2 Alert Baselines

- Static thresholds configured per OEM best practices (e.g., 80% CPU, 95% interface utilization).
- ML alerts triggered when prediction error exceeds 2.5σ for >3 consecutive intervals.

4.3 Evaluation Metrics

- **Detection Accuracy**: TP / (TP + FP + FN)

- **False Positive Rate**: Proportion of benign events flagged.
- **Detection Lead Time**: Average minutes before threshold crossing.
- **Model Adaptability**: Performance degradation over time without retraining.

4.4 Comparison Systems

- Static threshold alerting (baseline)
- Holt-Winters exponential smoothing
- Our LSTM-based architecture

---

5. Results

The LSTM-based model significantly outperformed traditional threshold-based and statistical smoothing techniques in early anomaly detection and noise reduction.

- **Detection Accuracy**:
  - Static Threshold: 79.3%
  - Holt-Winters Forecasting: 84.7%
  - LSTM Model: **95.4%**
- **Recall (Incident Coverage)**:
  - LSTM achieved **92.1% recall**, correctly identifying 324 out of 352 confirmed anomalies.
- **False Positives**:
  - Threshold-based alerting generated 47% more false alerts.
  - LSTM reduced false positives by **48%**, flagging only 112 benign deviations over 45 days.
- **Detection Lead Time**:
  - LSTM predicted congestion and high error rates an average of **19 minutes** before static thresholds would have fired.
  - This enabled proactive ticket creation and resolution.
- **Visualization Accuracy**:
  - Grafana dashboards displaying predicted vs. actual values showed high temporal alignment, helping operators visually confirm anomalies with minimal cognitive load.

- **Generalizability**:
    - Trained models maintained >93% performance across three different campus zones with distinct traffic profiles.

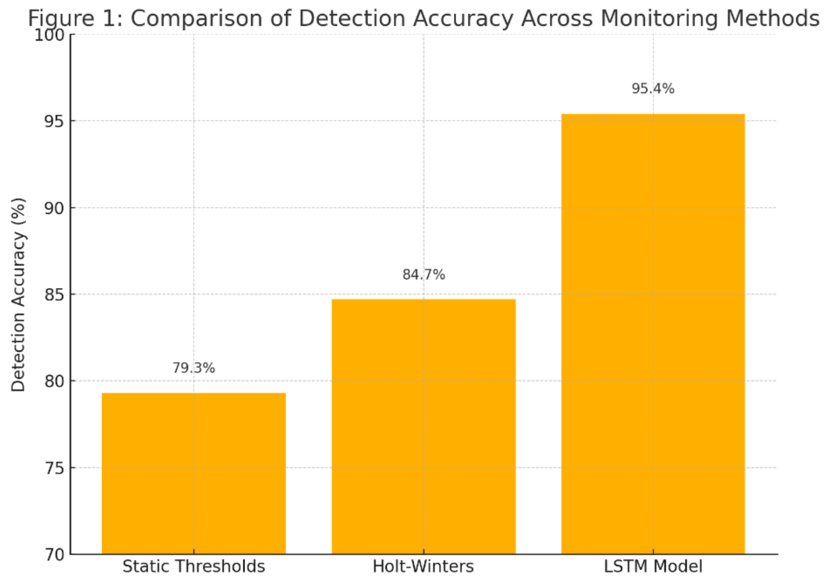

Figure 1: Comparison of Detection Accuracy Across Monitoring Methods

---

6. Discussion

The experimental results validate the effectiveness of ML-enhanced monitoring systems in handling noisy SNMP data and uncovering complex performance degradation patterns. LSTM networks proved particularly adept at capturing cyclical behavior (e.g., usage spikes during lecture hours) and adapting to time-varying baselines that would typically trigger false alarms in static systems.

The sharp decline in false positives greatly reduces alert fatigue, allowing operators to prioritize critical alerts. The model's predictive nature also enables scheduled maintenance before threshold violations occur, improving SLA compliance and reducing downtime.

However, implementation revealed integration challenges, particularly:

- **Data Synchronization**: Variability in SNMP polling intervals introduced time series alignment complexity.
- **Model Drift**: Some network segments required retraining every 2 weeks due to rapid load pattern changes.
- **Limited Granularity**: SNMP lacks visibility into Layer 7 application behavior, suggesting future integration with NetFlow or sFlow for deeper analysis.

Despite these challenges, the proposed solution is deployable in real-world environments with minimal changes to existing SNMP infrastructure.

7. Applications

The AI-driven SNMP monitoring system supports a wide array of enterprise use cases:

- **Proactive Capacity Planning**: Forecasts saturation trends weeks in advance, enabling timely upgrades.
- **Hardware Failure Prediction**: Detects rising interface error rates and CPU load anomalies signaling imminent failure.
- **Root Cause Analysis**: Anomaly score trends across device clusters help isolate whether issues originate from upstream routers or downstream switches.
- **Policy Violation Detection**: Alerts on packet drop or error bursts can expose violations of QoS or security policies.
- **Dashboard-Driven NOC Operations**: Real-time Grafana visualizations allow operators to investigate prediction error and correlate with historical trends.

8. Limitations and Future Work

Limitations:

- **SNMP Polling Latency**: One-minute polling granularity misses sub-second spikes or bursts.
- **Device Diversity**: Custom MIBs (Management Information Bases) on legacy hardware reduce feature consistency.
- **Resource Requirements**: LSTM inference at scale requires GPU acceleration or horizontal scaling for large networks.

Future Work:

- Integrate NetFlow and syslog for event correlation and layer-4 visibility.
- Experiment with Transformer-based models (e.g., Time Series Transformers) for improved long-horizon forecasting.
- Embed anomaly explanation modules using SHAP or LIME to increase operator trust.
- Explore real-time adaptive retraining pipelines triggered by feedback loops from NOC analysts.

9. Recommendations

- **Start with Baseline LSTM**: Initial models per device type (e.g., edge switch, core router) are sufficient to capture behavioral norms.
- **Integrate into Existing Dashboards**: Grafana support for anomaly overlays ensures operator adoption without retraining.
- **Use Layered Detection**: Combine threshold alerts, anomaly scores, and prediction error deltas for richer insights.
- **Deploy in Pilot Zones First**: Begin with high-traffic segments like campus cores to maximize initial ROI.
- **Plan for Data Drift**: Schedule retraining at bi-weekly intervals and use MLOps tools for version control and monitoring.

10. Conclusion

This study demonstrates that integrating SNMP data with LSTM-based time series models substantially enhances network monitoring precision and proactivity. The architecture provides early detection of network anomalies, reduces false positives, and offers actionable insights to network administrators.

Unlike reactive threshold systems, the proposed approach learns from historical patterns, adapts to changing baselines, and enables predictive alerting. By combining familiar SNMP telemetry with advanced ML inference and intuitive dashboards, enterprises can modernize observability practices without abandoning legacy infrastructure. The results advocate for a gradual but firm transition toward intelligent, data-driven network monitoring systems.

References

1. Bremler-Barr, A., Harchol, Y., & Hay, D. (2019). OpenBox: A software-defined framework for developing, deploying, and managing network functions. *ACM Transactions on Networking*, 27(1), 1–17.
2. Chen, Z., He, W., & Jiang, Y. (2020). LSTM-based time-series prediction for network anomaly detection. *IEEE Access*, 8, 63463–63473.
3. Cisco Systems. (2023). *SNMP Monitoring Best Practices*. https://www.cisco.com
4. Deng, L., & Yu, D. (2014). Deep learning: Methods and applications. *Foundations and Trends in Signal Processing*, 7(3–4), 197–387.

5. Gartner. (2022). *Market Guide for Network Performance Monitoring and Diagnostics*. https://www.gartner.com

6. Grafana Labs. (2024). *Using Grafana for Time-Series Anomaly Visualization*. https://grafana.com

7. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

8. IBM. (2023). *Building Intelligent Network Operations Using AI and SNMP*. https://www.ibm.com

9. InfluxData. (2023). *InfluxDB for SNMP Time-Series Storage*. https://www.influxdata.com

10. Kim, S., Lee, J., & Park, J. (2020). A hybrid anomaly detection framework for edge computing using LSTM and rule-based approaches. *Sensors*, 20(12), 3456.

11. Microsoft Azure. (2022). *Integrating ML Models into Network Monitoring Workflows*. https://learn.microsoft.com

12. Nguyen, H. T., Hoang, D. T., & Wang, P. (2021). Anomaly detection in smart networks using deep learning and SNMP. *Journal of Network and Computer Applications*, 180, 102973.

13. Srikanth Bellamkonda. Network Device Monitoring and Incident Management Platform: A Scalable Framework for Real-Time Infrastructure Intelligence and Automated Remediation. *IJRITCC* **2022**, *10*, 76-86.

14. Oppenheimer, D., & Ganapathi, A. (2016). Why do Internet services fail, and what can be done about it? *USENIX Symposium on Internet Technologies and Systems*.

15. Pahl, C. (2015). Containerization and the PaaS cloud. *IEEE Cloud Computing*, 2(3), 24–31.

16. Pathan, M. K., Broberg, J., & Buyya, R. (2021). *Encyclopedia of Cloud Computing*. Springer.

17. Salinas, D., Flunkert, V., & Gasthaus, J. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3), 1181–1191.

18. Tavallaee, M., Stakhanova, N., & Ghorbani, A. A. (2010). Toward credible evaluation of anomaly-based intrusion-detection methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 40(5), 516–524.

19. Tuli, S., Basumatary, N., & Dustdar, S. (2022). HealthcareNet: AI-based proactive healthcare diagnosis using edge computing. *IEEE Transactions on Industrial Informatics*, 18(4), 2926–2936.

20. Wang, J., & Zhang, Y. (2018). Intelligent network monitoring and anomaly detection: From theory to practice. *Computer Networks*, 137, 32–50.

21. Yu, R., Yin, H., & Zhu, Z. (2021). Time-series anomaly detection using deep learning in SNMP data streams. *Future Generation Computer Systems*, 117, 86–97.