МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЧЕРНІГІВСЬКА ПОЛІТЕХНІКА»
WIN-WIN EDIH



Co-funded by
the European Union



Співфінансується
Європейським Союзом



ВІЛЬНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ В ОСВІТІ, НАУЦІ ТА БІЗНЕСІ

FREE/OPEN SOURCE SOFTWARE IN EDUCATION, SCIENCE & BUSINESS (FOSS 2026)

Матеріали II Міжнародної науково-практичної конференції
01–02 жовтня 2026 р.
м. Чернігів

Чернігів 2026

УДК 004.49:[37+001+334.722]
В41

Рекомендовано до друку вченою радою Національного університету «Чернігівська політехніка» (протокол № 12 від 02.11.2026 р.)

Вільне програмне забезпечення в освіті, науці та бізнесі : матеріали ІІ Міжнар. В41 наук.-практ. конф. (м. Чернігів, 01–02 жовтня 2026 р.) / відп. за вип. Д. Б. Мехед. Чернігів : НУ «Чернігівська політехніка», 2026. 156 с.

ISBN 978-617-8584-09-2

Збірник містить матеріали ІІ Міжнародної науково-практичної конференції «Вільне програмне забезпечення в освіті, науці та бізнесі» (FOSS 2026), що відбулася онлайн 01–02 жовтня 2026 року в Національному університеті «Чернігівська політехніка». Розглянуто питання використання вільного та відкритого програмного забезпечення в освіті, науці, бізнесі, ІТ-інфраструктурі, кібербезпеці, аналізі даних і штучному інтелекті. Матеріали подано українською та англійською мовами.

Для науковців, викладачів, здобувачів освіти, фахівців ІТ-галузі та всіх, хто цікавиться вільним програмним забезпеченням.

Матеріали подано в авторській редакції. Відповідальність за зміст, достовірність фактів, цитат і посилань несуть автори.

Конференцію проведено в межах проєкту WIN-WIN EDIH (грантова угода № 101191647), співфінансованого Європейським Союзом у межах програми Digital Europe. Висловлені погляди та думки належать виключно автору(ам) і не обов'язково відображають позицію Європейського Союзу чи Європейської Комісії.

Відповідальний за випуск: Мехед Дмитро Борисович

ISBN 978-617-8584-09-2

© НУ «Чернігівська політехніка», 2026

Чопорова О. В., Чопоров С. В. ВИКОРИСТАННЯ 3D-МОДЕЛЮВАННЯ ТА АЛГОРИТМІВ НЕЙРОННИХ МЕРЕЖ В НАВЧАЛЬНИХ ПРОЄКТАХ ІНЖЕНЕРНО-ТЕХНІЧНОГО СПРЯМУВАННЯ / USE OF 3D MODELING AND NEURAL NETWORK ALGORITHMS IN EDUCATIONAL PROJECTS OF ENGINEERING AND TECHNICAL DIRECTION 82

Чуб М. В. ВИЯВЛЕННЯ ТА ПРОТИДІЯ АДАПТИВНИМ ВЕБ-АТАКАМ ЗА ДОПОМОГОЮ МОДЕЛЕЙ ГЛИБОКОГО НАВЧАННЯ 86

Шевченко Ю. О. ПОРІВНЯЛЬНЕ ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ POSTGRESQL ТА INFLUXDB ДЛЯ ЗБЕРІГАННЯ ДАНИХ ФІЗИЧНОЇ АКТИВНОСТІ / COMPARATIVE STUDY OF POSTGRESQL AND INFLUXDB PERFORMANCE FOR STORING PHYSICAL ACTIVITY DATA 89

Al-Akwaa K. H. M. A. OPEN-SOURCE DATA SCIENCE FOR SUSTAINABLE DEVELOPMENT: A REPRODUCIBLE ANALYTICAL FRAMEWORK 93

Borysenko O. A. AN ARCHITECTURAL APPROACH TO LLM-BASED ROOT CAUSE ANALYSIS FOR CONTAINER ORCHESTRATION INCIDENTS 100

Burmaka I. LOCAL, OPEN, AND ENERGY-EFFICIENT: HARNESSING X86 NPUS FOR EVERYDAY AI TASKS 102

Byvoino T. P., Byvoino P. G. SIMULATION MODELING AND STATISTICAL ANALYSIS OF THE EFFICIENCY OF AN EMERGENCY MEDICAL SERVICE STATION 106

Didusenko O. A., Bula Y. I., Shymchuk G. V. REPRODUCIBLE FOSS FRAMEWORK FOR RESILIENCE EVALUATION OF CONTAINERIZED DISTRIBUTED SYSTEMS 109

Didusenko O. A., Bula Y. I., Shymchuk G. V. REPRODUCIBLE FOSS TESTBED FOR FAULT-TOLERANT MULTIPATH ROUTING IN IP NETWORKS 112

Ebirim V., Jadama A. F. R. RESILIENT DIGITAL EDUCATION IN DISRUPTED ENVIRONMENTS: THE CCDS APPROACH 115

Fraya DIGITAL TRANSFORMATION OF BUSINESS THROUGH FREE AND OPEN SOURCE SOFTWARE: OPPORTUNITIES, BARRIERS, AND STRATEGIC PATHWAYS 119

Kala P. RUN MONGODB YOUR WAY: A PRACTICAL OPEN-SOURCE STACK FOR SELF-MANAGED DEPLOYMENTS 121

Lukin O. A MODERN C++ KMIP CLIENT BASED ON A REUSABLE PROTOCOL LIBRARY 124

Mekhed D. B. DEVELOPING STUDENTS' CYBER HYGIENE SKILLS THROUGH THE USE OF FREE SPECIALIZED SOFTWARE 127

Nahornyi B. I., Lysenok D. V., Savenko A. O. AN OPEN BROWSER-BASED CO-SIMULATOR OF STM32 FIRMWARE AND ELECTRICAL CIRCUITS AS A FREE ALTERNATIVE TO PROTEUS 130

Romanenko M. V. FOSS APPROACH IN MOBILE DEVELOPMENT: CREATING AN OPEN-SOURCE ANDROID APPLICATION WITH THE INTEGRATION OF READY-MADE LIBRARIES FOR STYLISTIC RECOMMENDATIONS 134

Shevchenko N. A., Kibitkin D. S., Shymchuk G. V. REPRODUCIBLE OPEN-SOURCE BENCHMARK OF CONCEPT-DRIFT DETECTORS FOR STREAMING REGRESSION 137

Shevchenko N. A., Kibitkin D. S., Shymchuk G. V. INTERPRETABLE OPEN-SOURCE MACHINE LEARNING FOR NATURAL GAS CONSUMPTION FORECASTING WITH WEATHER AND LAG FEATURES 140

Shivashankar C. ARCHITECTING DATABASE DEPLOYMENTS ON KUBERNETES: PATTERNS, TRADE-OFFS AND EXPERIENCES 143

Shymchuk G. V., Yatskiv P. O. DETECTION OF ANOMALOUS STATES IN IP NETWORKS USING MACHINE LEARNING AND OPEN-SOURCE SOFTWARE TOOLS 146

Shymchuk G. V., Yatskiv P. O. OPEN-SOURCE SOFTWARE ENVIRONMENT FOR MODELING FAULT-TOLERANT MULTIPATH ROUTING IN IP NETWORKS 149

Silva F. HOSTING SOMEONE ELSE'S APP: IFRAME CHOREOGRAPHY IN PMM 152

Šupić A. ZERO-DOWNTIME MIGRATION WITH PERCONA CLUSTERSYNC FOR MONGODB 154

Shymchuk G. V.1, Yatskiv P. O.2
1 – Ternopil Ivan Puluj National Technical University, Department of Computer Science, Ternopil, Ukraine
2 – Ternopil Ivan Puluj National Technical University, Department of Computer-Integrated Technologies, Ternopil, Ukraine
Corresponding author: Grigorii Shymchuk, e-mail: gorych@gmail.com, ORCID: 0000-0003-2362-7386

DETECTION OF ANOMALOUS STATES IN IP NETWORKS USING MACHINE LEARNING AND OPEN-SOURCE SOFTWARE TOOLS

Abstract. An approach to detecting anomalous states in IP networks from network telemetry using open-source machine-learning tools is investigated. A reproducible synthetic dataset was generated with six indicators: throughput, RTT, jitter, packet loss, retransmissions, and link utilization. Isolation Forest, Local Outlier Factor, and One-Class SVM were trained using only normal observations and then compared on a mixed test set. For 900 test observations, Local Outlier Factor achieved the highest F1 = 0.9560 and accuracy = 0.9700; Isolation Forest obtained F1 = 0.9472, while One-Class SVM obtained F1 = 0.9346. The pipeline is implemented with Python and scikit-learn and can be integrated with Mininet/iperf3 for subsequent validation under reproducible emulated network scenarios.

Keywords: open-source software; IP networks; anomalies; machine learning; Isolation Forest; Local Outlier Factor; One-Class SVM.

Introduction

The increasing complexity of IP networks, distributed systems, and cloud infrastructures raises the requirements for automated state monitoring. Operational anomalies may appear as sudden growth in delay, jitter, and packet loss, throughput degradation, link congestion, or route instability. Such changes are not necessarily caused by cyberattacks, but they shift telemetry away from the normal operating profile and may precede service degradation.

Recent network-anomaly studies actively use unsupervised learning because comprehensive traffic labeling is expensive and the complete set of abnormal states is rarely known in advance. Combined unsupervised methods for near-real-time network traffic analysis were investigated in [1], while dynamic network traffic profiling was studied in [2]. Comparative research indicates that the effectiveness of Isolation Forest, LOF, and One-Class SVM depends on data structure and model configuration [3, 5]. More complex unsupervised architectures based on Transformers and autoencoders are also used in modern network systems [4]; however, classical methods remain attractive for lightweight FOSS solutions because they are easier to deploy and computationally less demanding.

The objective of this work is to experimentally compare three unsupervised machine-learning methods for detecting anomalous IP-network states from operational indicators and to construct a reproducible open-source pipeline suitable for subsequent integration with an emulated network testbed.

Materials and Methods

The software pipeline was implemented in Python using NumPy/pandas for data generation and processing and scikit-learn 1.9.1 for model training. Scikit-learn 1.9.1 was released in September 2026 [7]. Future validation on a network testbed is designed to use Mininet/Open vSwitch for topology emulation and iperf3 3.21 for generating TCP/UDP load and measuring throughput [8]. The overall experimental workflow is shown in figure 1.

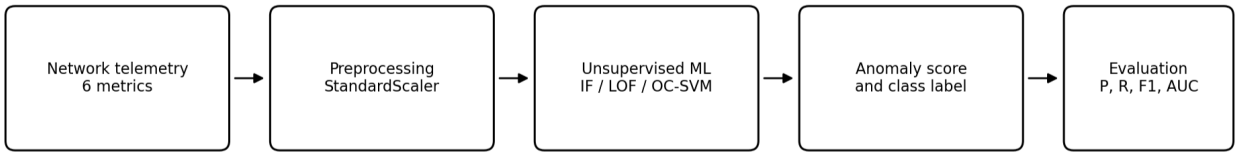

Figure. 1 – Reproducible open-source network-anomaly detection pipeline

To separate algorithm behavior from uncontrolled properties of real traffic, the first stage used a controlled synthetic telemetry dataset generated with a fixed random seed of 42. Each observation contained six numerical features: throughput (Mbit/s), RTT (ms), jitter (ms), packet loss (%),

retransmission count, and utilization (%). The training set contained 1,800 normal observations. The test set consisted of 600 normal and 300 anomalous observations. The anomalous samples were distributed among four scenarios: link congestion, packet-loss burst, abnormally high latency, and route instability.

Table 1 – Scenarios used to form anomalous states

State	Throughput / utilization	Delay	Jitter / loss	Interpretation
Normal	Stable	Stable	Low	Baseline profile
Congestion	↓ throughput, ↑ utilization	↑ RTT	↑ jitter/loss	Link degradation
Loss burst	Moderate ↓ throughput	Moderate ↑ RTT	Sharp ↑ loss/retrans.	Packet anomaly
High latency	Near-normal throughput	Sharp ↑ RTT	↑ jitter	Latency anomaly
Route instability	Throughput fluctuations	↑ RTT	↑ jitter/loss	Route instability

Before training, all features were standardized using parameters estimated only from the training set. Three models were compared: Isolation Forest with 300 trees and contamination = 0.03; Local Outlier Factor with n_neighbors = 35, contamination = 0.03, and novelty = True; and One-Class SVM with an RBF kernel, nu = 0.03, and gamma = scale. All models were trained exclusively on normal telemetry, anomaly labels were used only for final evaluation.

Accuracy, precision, recall, F1, and ROC-AUC were used for evaluation. F1 was selected as the primary metric because it simultaneously reflects the fraction of correctly detected anomalies and the number of false alarms. For the anomaly class: Precision = TP/(TP+FP), Recall = TP/(TP+FN), and F1 = 2·Precision·Recall/(Precision+Recall).

Results

The experimental results are presented in Table 2. All three algorithms achieved high ROC-AUC values (>0.99), confirming their ability to rank most abnormal states away from the normal baseline. At the same time, the operating decision boundary substantially affected the number of false alarms.

Table 2 – Anomaly-detection results

Model	Accuracy	Precision	Recall	F1	ROC-AUC
Isolation Forest	0.9644	0.9379	0.9567	0.9472	0.9932
Local Outlier Factor	0.9700	0.9361	0.9767	0.9560	0.9945
One-Class SVM	0.9544	0.8960	0.9767	0.9346	0.9948

Local Outlier Factor provided the best balance between precision and recall: accuracy = 0.9700, recall = 0.9767, and F1 = 0.9560. It missed 7 of 300 anomalous observations and produced 20 false alarms among 600 normal observations. Isolation Forest had slightly higher precision = 0.9379 but lower recall = 0.9567, resulting in F1 = 0.9472. One-Class SVM reached recall = 0.9767 and the highest ROC-AUC = 0.9948, but produced 34 false positives; consequently, precision decreased to 0.8960 and F1 to 0.9346.

Discussion

These findings are consistent with recent studies emphasizing that unsupervised detector performance strongly depends on data characteristics and hyperparameter settings [3, 5]. LOF performed best in the present experiment because the simulated abnormal states formed local low-density regions relative to the normal operating profile. Isolation Forest remained competitive and offers a simple interpretation through isolation of atypical observations. One-Class SVM provided strong ranking according to ROC-AUC but requires more careful tuning of the decision boundary or nu to reduce the false-positive rate.

A practical advantage of the proposed approach is its fully open-source stack: telemetry can be collected with iperf3 and operating-system utilities, data can be prepared in Python, and detection can be implemented with scikit-learn. This orientation is consistent with recent open-source network anomaly detection prototypes integrating Isolation Forest with open monitoring platforms

and Python processing [6]. The pipeline is therefore suitable for laboratory and educational testbeds and for model preselection before more complex deployment.

The main limitation is the synthetic nature of the telemetry; the absolute metric values obtained here should not be transferred directly to production networks. The experiment primarily validates the comparison methodology and reproducibility. The next stage should reproduce the same scenarios in Mininet with real TCP/UDP iperf3 flows, collect time-series telemetry at identical sampling intervals, and evaluate stability across several topologies and load levels.

Conclusions

A reproducible open-source pipeline for detecting anomalous IP-network states from six telemetry indicators was developed. Isolation Forest, Local Outlier Factor, and One-Class SVM were compared on a synthetic test set containing 900 observations. Local Outlier Factor achieved the highest F1 = 0.9560 and accuracy = 0.9700; Isolation Forest achieved F1 = 0.9472, and One-Class SVM achieved F1 = 0.9346.

The results support the use of lightweight unsupervised algorithms as a baseline intelligent monitoring layer for IP networks. Future work will transfer the experiment from synthetic telemetry to Mininet/iperf3, analyze temporal anomaly sequences, and test adaptive decision thresholds to reduce false alarms.

References

1. Carrera F., Dentamaro V., Galantucci S., Iannacone A., Impedovo D., Pirlo G. Combining Unsupervised Approaches for Near Real-Time Network Traffic Anomaly Detection. Applied Sciences, 2022, 12(3), 1759. https://doi.org/10.3390/app12031759.
2. Yi J., Zhang S., Tan L., Tian Y. A Network Traffic Abnormal Detection Method: Sketch-Based Profile Evolution. Applied Sciences, 2023, 13(16), 9087. https://doi.org/10.3390/app13169087.
3. Agyemang E.F. Anomaly Detection Using Unsupervised Machine Learning Algorithms: A Simulation Study. Scientific African, 2024, 26, e02386. https://doi.org/10.1016/j.sciaf.2024.e02386.
4. Geng H., Ma Q., Chi H., Zhang Z., Yang J., Yin X. DUdetector: A Dual-Granularity Unsupervised Model for Network Anomaly Detection. Computer Networks, 2025, 257, 110937. https://doi.org/10.1016/j.comnet.2024.110937.
5. Iturbe-Araya J.I., Rifà-Pous H. Enhancing Unsupervised Anomaly-Based Cyberattacks Detection in Smart Homes through Hyperparameter Optimization. International Journal of Information Security, 2025, 24, 45. https://doi.org/10.1007/s10207-024-00961-6.
6. Ivanova V., Genkov D. A Machine Learning-Based Network Anomaly Detection System Prototype Using Isolation Forest. Engineering Proceedings, 2026, 154(1), 7. https://doi.org/10.3390/engproc2026154007.
7. scikit-learn Developers. scikit-learn 1.9.1 Release, September 2026. https://scikit-learn.org/stable/whats_new/v1.9.html (accessed 10.09.2026).
8. ESnet. iperf3 Project News: iperf 3.21 Released, 9 April 2026. https://software.es.net/iperf/news.html (accessed 10.09.2026).