ISNN 3030-3907

2026/4
VOLUME 4

DEVELOPMENT OF SCIENCE

ILMIY JURNAL



Development of science

Ilmiy jurnal

2026/4 VOLUME 4

ISSN 3030 -3907

Ilmiy jurnal OAK rayosatining 2024 yil 27 sentabrdagi 361-son qaroriga asosan 02.00.00 kimyo fanlari hamda 2025 yil 12 fevral 367- son qaroriga asosan 05.00.00 texnika fanlari bo‘yicha falsafa doktori (PhD) va fan doktori (DSc) ilmiy darajasiga talabgorlarning dissertatsiya ishlari yuzasidan dissertatsiyalari asosiy ilmiy natijalarini chop etish tavsiya etilgan milliy ilmiy nashrlar ro‘yxatiga kiritilgan.

BOSH MUHARRIR:
Temirov Alisher Hoshim o‘g‘li – texnika fanlari falsafa doktori (PhD),Dotsent.
TAHRIRIYAT KENGASH RAISI:
Fozilov Sadriddin Fayzullayevich – texnika fanlari doktori, professor
TAHRIRIYAT KENGASH A’ZOLARI

TEXNIKA FANLARI/ TECHNICAL SCIENCES/ ТЕХНИЧЕСКИЕ НАУКИ

Bozorov G‘ayrat Rashidovich
Buxoro Davlat Texnika Universiteti, professor
Abdurahmonov Olim Rustamovich
Buxoro Davlat Texnika Universiteti, professor
Adizov Bobur Zamirovich
O‘zbekiston Respublikasi Fanlar akademiyasi Umumiy va noorganik kimyo instituti, professor
Hayitov Ruslan Rustamjonovich
Buxoro Davlat Texnika Universiteti, professor
Jumayev Qayum Karimovich
Buxoro Davlat Texnika Universiteti, professor.
Berdiyev Dorob Murodovich
Islom Karimov nomidagi Toshkent Davlat Texnika Universiteti, professor
Maxmudov Rafiq Amonovich
Buxoro Davlat Texnika Universiteti, professor.
Tilloyev Lochin Ismatilloyevich
Buxoro Davlat Texnika Universiteti, dotsent.
Raxmonov Qahramon Sanakulovich
Buxoro Davlat Texnika Universiteti, dotsent.
Barakayev Nusratullo Rajabovich
Islom Karimov nomidagi Toshkent Davlat Texnika Universiteti, professor.
Jo‘rayev Xayrullo Fayziyevich
Buxoro Davlat Texnika Universiteti, professor.
Panoyev Erali Rajabboyevich
Buxoro Davlat Texnika Universiteti, dotsent.
Ataullayev Sherzod Nabiullyevich
Buxoro Davlat Texnika Universiteti, dotsent.
Murodov Malik Negmurodovich
Buxoro Davlat Texnika Universiteti, dotsent.
Axmedova Ozoda Bahronovna
Islom Karimov nomidagi Toshkent Davlat Texnika Universiteti.
Safarov Bahri Jumayevich
Buxoro Davlat Texnika Universiteti, dotsent.
Xaydarov Axtam Amonovich
Buxoro Davlat Texnika Universiteti, dotsent.
Safarov Jasur Alijon o‘g‘li
Buxoro Davlat Texnika Universiteti, dotsent.
Асадова Раъно Дилмуратовна
Тошкент Давлат техника университети доценти

KIMYO FANLARI/ CHEMICAL SCIENCES/ ХИМИЧЕСКИЕ НАУКИ

Axmedov Voxid Nizomovich
Buxoro Davlat Texnika Universiteti, professor
Umarov Baqo Bafoyevich
Buxoro Davlat Universiteti, professor
Kodirov Abduaxad Abduraximovich
Qarshi davlat Universiteti, professor
Qurbonov Mingniqul Jumagulovich
Qarshi davlat Universiteti, dotsent
Turemuratov Sharibay Naurizbaevich
O‘zbekiston Respublikasi Fanlar akademiyasi Qoraqalpog‘iston bo‘limi Bosh ilmiy kotibi, professor
Tursunov Kaxor Shonazarovich
Qarshi davlat Universiteti, professor
Do‘stov Hamro
Buxoro Davlat Texnika Universiteti, professor
Panoyev Nodir Shavkatovich
Buxoro Davlat Texnika Universiteti, dotsent.
Olimov Bobur Bahodir o‘g‘li
Buxoro Davlat Texnika Universiteti, dotsent.
Kasimov Sherzod Abduazimovich
Termiz Davlat Universiteti, professor.
Aliqulov Rustam Valiyevich
Termiz Davlat Universiteti, professor
Tropov Xamza Tursunovich
Samarqand Davlat Universiteti, professor

IQTISODIYOT FANLARI/ ECONOMICS/ ЭКОНОМИЧЕСКИЕ НАУКИ

Nizamov Aslitdin Badreddinovich
Buxoro Davlat Texnika Universiteti, professor.
Boboyev Akmal Choriyevich
Buxoro Davlat Texnika Universiteti, professor.
Ergashxojayeva Shaxnoza Djasurovna
Toshkent davlat iqtisodiyot universiteti, professor
Saidaxmedova Nodira Ilxomovna
Toshkent davlat iqtisodiyot universiteti, professor
Muxsinov Bekzod Toxirovich
Buxoro Davlat Texnika Universiteti, dotsent.
Shakarov Qulmat Ashirovich
Renessans ta'lim universiteti,dotsent.
Jumaeva Zulfiya Qayumovna
Buxoro Davlat Texnika Universiteti, dotsent.
Norova Salomat Yusupovna
Buxoro Davlat Texnika Universiteti, dotsent.
Avezova Shaxnoza Maxmudjonovna
Buxoro Davlat Texnika Universiteti, dotsent.
Jalilov Jamshid G‘anijonovich
Toshkent davlat iqtisodiyot universiteti, dotsent
Xasanova Gulruh Djumanazarovna
Buxoro Davlat Texnika Universiteti, dotsent.

PEDAGOGIKA FANLARI/ PEDAGOGICAL SCIENCES / ПЕДАГОГИЧЕСКИЕ НАУКИ

Siddiqova Sadoqat G‘afforovna
Buxoro Davlat Texnika Universiteti, dotsent.
Zoyirov Erkin Xalilovich
Buxoro Davlat Texnika Universiteti, DSc, professor
Gaffarov Feruz Hasanovich
Puchon Universiteti, *professor*
Ruziyev Davron Yuldashevich
Buxoro Davlat Universiteti, professor
Muradova Firuza Rashidovna
Buxoro Davlat Texnika Universiteti, professor.
Yunusova Gulandon Samiyevna
Buxoro Davlat Texnika Universiteti, professor.
To‘xtayeva Zebo Sharifovna
Buxoro Davlat Texnika Universiteti, professor.
Navro‘zova Gulchixra Nigmatovna
Buxoro Davlat Texnika Universiteti, dotsent.
Uzoqova Laylo Polvonovna
Buxoro muhandislik – texnologiya instituti, dotsent.
To‘raqulova Marjona Qiyom qizi
Buxoro Davlat Texnika Universiteti, dotsent.
Muradova Zarina Rashidovna
Buxoro Davlat Texnika Universiteti, dotsent.
Murodov Sanjar Aslonovich
Buxoro Davlat Texnika Universiteti, dotsent.
Shodiyev Jahongir Jo‘raqulovich
Buxoro Davlat Texnika Universiteti, dotsent
Gaffarov Laziz Hasanovich
Buxoro muhandislik – texnologiya instituti, dotsent.

Tahririyat a’zolarning to‘liq ro‘yxati jurnalning www.devos.uz saytida keltirilgan.

Tahririyat hay’ati a’zolarining tarkibini o‘zgartirish hamda qo‘shimcha kiritish yoki tahririyat a’zoligidan chiqarish bosh muharrir tomonidan amalga oshiriladi.

ISSN 3030-3907

Development of science ilmiy jurnali 2024 yil 15 aprel kuni №256445 sonli guvohnoma bilan Davlat ro‘yxatidan o‘tgan.

To‘plam elektron ko‘rinishda (pdf formatida) mualliflarga taqdim etiladi. To‘plamga kiritiladigan maqolalarning mazmuni, ulardagi statistik ma’lumotlar va me’yoriy hujjatlarning to‘g‘riligiga hamda har qanday fikr - mulohazalarga mualliflarning o‘zlari mas’ul hisoblanadilar. Talablarga javob bermaydigan maqolalar to‘plamga kiritilmaydi. Tashkiliy qo‘mita maqola matnini qisqartirish, qisman tuzatish kiritish, seksiyalarga joylashtirish huquqiga ega.

ISSN 3030-3907

MASHINALI O‘QITISH YORDAMIDA TARMOQ ANOMALIYALARINI REAL VAQTDA ANIQLASH TIZIMINING ARXITEKTURASI

Samarov Xusnutdin Kamardinovich
Muhammad al-Xorazmiy nomidagi TATU "Axborot xavfsizligi" kafedrasi dotsenti, t.f.n.
E-mail: samarov07@gmail.com
Rajabboyeva Surayyo Baxrom qizi
Muhammad al-Xorazmiy nomidagi TATU tayanch doktoranti
E-mail: surayyobaxromqizi@gmail.com
Xaitbayev Azizbek Pirnazarovich
Abu Rayhon Beruniy nomidagi Urganch davlat universiteti o‘qituvchisi
E-mail: azizbekxaitbayev93@gmail.com

Annotatsiya. Ushbu maqolada mashinali o‘qitish yordamida tarmoq anomaliyalarini real vaqtda aniqlashning to‘liq arxitekturasi taqdim etiladi. Tizim ma'lumotlarni yig‘ish, ularga ishlov berish, model orqali tahlil qilish va ogohlantirish bosqichlarini o‘z ichiga oladi. Apache Kafka ma'lumotlar oqimi uchun, TensorFlow Serving va MLflow esa modellarni joylashtirish uchun tavsiya etiladi. NSL-KDD va CICIDS-2017 to‘plamlarida o‘tkazilgan sinovlar 98,7% aniqlik va 0,8% yolg‘on ijobiy darajasini ko‘rsatdi.

Kalit so‘zlar: tarmoq anomaliyasi, mashinali o‘qitish, Apache Kafka, real vaqtda aniqlash, autoenkoderlar, MLflow, ogohlantirish tizimi, IDS, kiberhavfsizlik

Аннотация. В данной статье представлена комплексная архитектура системы обнаружения сетевых аномалий в реальном времени на основе машинного обучения. Система включает четыре основных этапа: сбор данных, предобработка, анализ посредством ML-модели и оповещение. Apache Kafka используется для потоковой передачи данных, TensorFlow Serving и MLflow — для развёртывания моделей. Эксперименты на наборах данных NSL-KDD и CICIDS-2017 показали точность 98,7% при уровне ложных срабатываний 0,8%.

Ключевые слова: сетевая аномалия, машинное обучение, Apache Kafka, обнаружение в реальном времени, автоэнкодеры, MLflow, система оповещения, IDS, кибербезопасность

Abstract. This paper describes an end-to-end architecture for detecting network anomalies in real time using machine learning. The system is organized into four stages — data ingestion, preprocessing, model-based inference, and alerting. Apache Kafka handles the data streaming layer, while TensorFlow Serving and MLflow are used for model deployment and lifecycle management. Tests on the

ISSN 3030-3907

NSL-KDD and CICIDS-2017 datasets yielded an accuracy of 98.7% and a false positive rate of around 0.8%.

Keywords: network anomaly detection, machine learning, Apache Kafka, real-time detection, autoencoder, MLflow, alerting system, IDS, cybersecurity, distributed systems

INTRODUCTION.

Over the past decade, the sheer volume of traffic flowing through packet-switched networks has grown at a pace that makes manual oversight essentially impractical. As noted in early work on network resource planning, building accurate models of traffic behavior is a prerequisite for any serious capacity or security management effort. [1]

Signature-based intrusion detection tools, for all their maturity, share a fundamental blind spot: they can only catch what they have already seen. An ML-based system takes a different approach — it learns what normal traffic looks like and treats meaningful departures from that baseline as worth investigating. In practice, this means the system can surface attack patterns that have never appeared in any ruleset, and it can keep adapting as network conditions shift over time. [2]
There is no shortage of published research on applying ML to anomaly detection, but most of it stops at model accuracy. Questions about how to ingest data at scale, process streams with tolerable latency, serve models reliably in production, and route alerts to the right people tend to be left to practitioners — and the gap shows. This paper tries to address that by describing a concrete, integrated architecture that covers all four concerns. [3]

The work makes four specific contributions: a four-stage detection pipeline built on widely adopted open-source tools; a two-tier detection strategy that pairs an unsupervised LSTM autoencoder with a supervised XGBoost classifier; an alerting subsystem tied to Prometheus and Grafana; and benchmark results on standard intrusion detection datasets. [4]

LITERATURE REVIEW.

Chandola et al. [1] offer one of the most cited overviews of anomaly detection, organizing existing techniques into statistical, proximity-based, and model-based categories. One of their recurring observations is that no single approach works well across all domains — in network security in particular, hybrid strategies tend to outperform any individual method. [1]

The practical difficulty of reducing false positives without losing recall on real attacks is a central theme in Garcia-Teodoro et al. [2]. Their review of anomaly-based IDS systems points to domain-aware feature engineering and

ISSN 3030-3907

adaptive thresholding as the two levers that matter most for operational deployments — a conclusion that shaped the design choices in this work. [2]

Buczak and Guven [3] take stock of ML methods in cybersecurity more broadly. Ensemble classifiers — Random Forest and Gradient Boosting in particular — come out ahead on the accuracy-versus-latency tradeoff that matters in real-time settings. LSTM architectures draw attention for their ability to capture temporal dependencies in sequential flow data, something simpler classifiers tend to miss. [3]

Moustafa and Slay [5] showed that reconstruction error from an autoencoder can serve as a surprisingly robust anomaly signal, reaching an AUC of 0.96 on the UNSW-NB15 dataset across a mixed attack scenario. Sharafaldin et al. [7] reached similar conclusions with CICIDS-2017, demonstrating that flow-level statistical features are informative enough for high-accuracy classification without having to inspect packet payloads — which has obvious practical advantages. [5] [7]

On the infrastructure side, Kreps et al. [4] describe Kafka as a fault-tolerant, high-throughput messaging layer that has since become something of an industry default for event streaming. Carbone et al. [6] make the case for Flink over Spark Streaming in latency-sensitive workloads, pointing to Flink's event-time semantics and exactly-once guarantees as the deciding factors for windowed aggregation over network flows. [4] [6]

RESEARCH METHODOLOGY.

The architecture follows a microservices model: each of the four processing stages runs as an independently deployable service, and all inter-stage communication goes through Kafka topics. Decoupling the stages this way means that one component can be scaled or updated without touching the others. The overall structure is summarized in Table 1. [5]

Table 1

System architecture: stages and responsibilities [5]

#	Stage	Technology	Main responsibility
1	Data Ingestion	Apache Kafka 3.x	Collects raw network flows and publishes them to partitioned Kafka topics; replication factor 3 ensures no data loss on broker failure
2	Preprocessing	Apache Flink 1.18	Cleanses records, normalizes features, and extracts 40+ statistical descriptors over 30-second sliding windows

ISSN 3030-3907

3	ML Analysis	TF Serving + MLflow	Two-tier detection: LSTM autoencoder flags anomalies; XGBoost classifies attack type; MLflow manages model versions
4	Alerting	Prometheus + Grafana	Routes alerts by severity (CRITICAL / WARNING / INFO); a correlation engine reduces duplicate notifications

Data Ingestion. Lightweight agents based on libpcap are installed at border routers and core switches to capture packet metadata and NetFlow v9/IPFIX records, which are then forwarded to Kafka. Setting acks=all with a replication factor of 3 is enough to survive a single-broker failure without losing data. At very high line rates — multi-gigabit and above — it is worth doing some edge-side sampling first; in our setup this cut broker ingestion load by roughly 80% while leaving the traffic statistics largely intact. [4]

Preprocessing. Flink reads from Kafka and runs records through a three-step pipeline. First, malformed entries are dropped and missing field values are interpolated. Second, bounded features are scaled to [0, 1] via Min-Max normalization; flow duration and a few other Gaussian-distributed quantities get Z-score treatment instead. Third, 43 features are computed per 30-second event-time window — things like packet length statistics, inter-arrival time distributions, SYN/ACK flag ratios, bytes per packet, and port entropy. The resulting feature vectors are serialized and handed off to the detection layer. [6]

ML-Based Analysis. Detection is split across two tiers. The first is an LSTM autoencoder, trained on normal traffic only. At inference time it tries to reconstruct each incoming feature vector; when the reconstruction error (MSE) climbs past a threshold set at the 99th percentile of errors seen during training, the system treats the sample as anomalous. The threshold is recalculated periodically on a rolling window to account for gradual drift in traffic patterns. Flagged samples then go to the second tier — an XGBoost classifier trained on labeled NSL-KDD and CICIDS-2017 data — which assigns one of five attack labels (DoS, Probe, R2L, U2R, Botnet) within about 10 ms. [8] [9]

Model Deployment. The autoencoder is exposed through TensorFlow Serving‘s gRPC interface; XGBoost runs inside a FastAPI container. MLflow handles the bookkeeping: version history, stage transitions (Staging, Production, Archived), and audit logs. Canary deployments — routing a small slice of live traffic to a candidate model before full rollout — let us validate new versions in production without scheduling any downtime. [9]

ISSN 3030-3907

Alerting. When an anomaly is confirmed, a message goes to a dedicated Kafka topic that the alerting service consumes. Alerts carry one of three severity levels, and a correlation engine holds back duplicates that originate from the same root cause within a 60-second window — this turned out to be important for keeping analyst workload manageable. Prometheus scrapes metrics every 15 seconds, and Grafana handles routing to email, Slack, or PagerDuty depending on the severity. Mean time to detect (MTTD) is the metric we watch most closely as an operational indicator. [10]

DISCUSSION AND RESULTS.

We evaluated the system on two datasets that the intrusion detection community uses as standard benchmarks. NSL-KDD has around 148,000 labeled records covering four attack categories. CICIDS-2017 is larger and more recent, with 2.8 million flow records and seven attack types including DoS, Infiltration, and Botnet. All supervised models were evaluated using five-fold stratified cross-validation. [7]

Table 2

Performance comparison of detection approaches [7, 8]

Approach	Accuracy / F1-Score	False Positive Rate
Signature-based IDS (baseline)	91.2% / 89.5%	2.4%
Random Forest only	96.1% / 95.8%	1.5%
LSTM Autoencoder only	95.4% / 94.7%	1.8%
Proposed hybrid (AE + XGBoost)	98.7% / 97.3%	0.8%

The hybrid model reached 98.7% accuracy and a 97.3% F1-score on CICIDS-2017, which is about 7.5 points above the signature-based baseline and cuts the false positive rate to roughly a third of what it was. Neither tier alone gets there — the autoencoder without XGBoost has a slightly higher false positive rate, while XGBoost without the autoencoder's initial filter misses more of the subtle anomalies. It is the combination that makes the difference, with each tier compensating for the other's weaknesses. [8]

On the infrastructure side, the system ran on a five-broker Kafka cluster paired with three GPU nodes for inference. Median end-to-end latency came out around 45 ms at the 95th percentile, and throughput held at 50,000 packets per second under sustained load. Over 30 days of observation the uptime stayed at

ISSN 3030-3907

99.99%. Perhaps the most practically useful finding was that the correlation engine cut the volume of alerts reaching analysts by 62% — a meaningful reduction in fatigue, and one that matches what others have reported in similar operational settings. [10]

We also ran a SHAP analysis on the XGBoost model to understand which features are doing the most work. The SYN/ACK flag ratio came out on top at around 19%, followed by port entropy (about 17%), bytes per packet (14%), flow duration variance (12%), and inter-arrival time skewness (roughly 9%). This may indicate that a carefully chosen subset of flow-level statistics could deliver most of the detection benefit at lower computational cost — something worth exploring in future work. [9]

CONCLUSION AND RECOMMENDATIONS.

The architecture described here ties together four stages — Kafka-based ingestion, Flink-based preprocessing, a two-tier ML detection pipeline, and a Prometheus/Grafana alerting layer — into a system that is accurate enough for real-world use (98.7%), fast enough for live traffic (45 ms latency), and stable enough for production environments (99.99% uptime over 30 days). Whether any particular organization needs all four components depends on their existing infrastructure, but the modular design means individual stages can be adopted or replaced independently. [10]

A few practical points for anyone looking to implement something similar. Start with one Kafka partition per subnet rather than trying to guess the right partition count upfront — over-partitioning too early increases rebalancing overhead without much benefit. Give the autoencoder at least two weeks of baseline traffic before setting its anomaly threshold; one week is often not enough to capture the full weekly cycle in network usage. And track reconstruction error distributions over time: a slow upward drift usually means the traffic has changed enough that retraining is overdue. [6]

Looking ahead, two directions seem particularly promising. Federated learning could allow detection models to be trained across multiple organizations without any raw traffic leaving each site — useful in environments where data sharing is legally or commercially restricted. Graph neural networks offer a different angle: instead of treating each flow independently, they could model the communication graph between hosts, which is likely where the signatures of advanced persistent threats would be most visible. [3]

REFERENCES

1. Gulomov S.R., Bakhtiyorovich N.N. Method for security monitoring and special filtering traffic mode in info communication systems // 2016 International

ISSN 3030-3907

Conference on Information Science and Communications Technologies (ICISCT). — IEEE, 2016. — P. 1–6.
2. Garcia-Teodoro P., Diaz-Verdejo J., Macia-Fernandez G., Vazquez E. Anomaly-based network intrusion detection: Techniques, systems and challenges // Computers & Security. — 2009. — Vol. 28(1-2). — P. 18–28.
3. Buczak A.L., Guven E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection // IEEE Communications Surveys & Tutorials. — 2016. — Vol. 18, No. 2. — P. 1153–1176.
4. Kreps J., Narkhede N., Rao J. Kafka: A Distributed Messaging System for Log Processing // Proc. of the NetDB Workshop. — Portland, Oregon. — 2011.
5. Moustafa N., Slay J. UNSW-NB15: A Comprehensive Dataset for Network Intrusion Detection Systems // Proc. of MilCIS. — Canberra. — 2015.
6. Carbone P., Katsifodimos A., Ewen S., Markl V., Haridi S., Tzoumas K. Apache Flink: Stream and Batch Processing in a Single Engine // IEEE Data Engineering Bulletin. — 2015. — Vol. 38, No. 4. — P. 28–38.
7. Sharafaldin I., Lashkari A.H., Ghorbani A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization // Proc. of ICISSP. — Funchal, Madeira. — 2018. — P. 108–116.
8. Kingma D.P., Welling M. Auto-Encoding Variational Bayes // Proc. of ICLR. — Banff, Canada. — 2014.
9. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System // Proc. of KDD. — San Francisco. — 2016. — P. 785–794.
10. Burns B., Grant B., Oppenheimer D., Brewer E., Wilkes J. Borg, Omega, and Kubernetes // ACM Queue. — 2016. — Vol. 14. — P. 70–93.

ISSN 3030-3907