INTERNATIONAL JOURNAL OF

INNOVATIONS IN APPLIED SCIENCE

AND ENGINEERING

e-ISSN: 2454-9258; p-ISSN: 2454-809X

Self-Supervised Learning for Cybersecurity Anomaly Detection

Seena K R

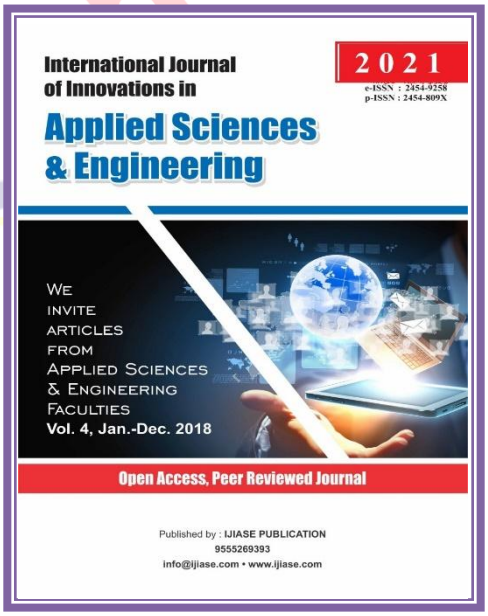
Lecturer in Computer Engineering
Government Polytechnic College Palakkad, Kerala, India

Paper Received: 24th June 2021; **Paper Accepted:** 09th July 2021;

Paper Published: 03rd October 2021

How to cite the article:

Self-Supervised Learning for Cybersecurity Anomaly Detection, IJIASE, January-December 2021, Vol 7, Issue: 1; 295-308



ABSTRACT

Cybersecurity systems deal with a huge amount of data like network traffic, system logs, user login records, and information from devices. Because of this, it's hard to monitor everything by hand or use old rules to catch security threats. Newer systems that use machine learning have shown promise, but most of them rely on labeled data, which is costly to create, quickly becomes outdated, and doesn't cover new or unknown attacks well. Self-supervised learning offers a better way. It learns from large amounts of data that hasn't been labeled, using automatically created learning goals. Recent studies show that contrastive self-supervised learning can find unusual activity in encrypted network traffic without looking at the actual content of the packets. This research introduces a new framework called SS-CADF, which uses unlabeled traffic data to learn what normal and abnormal network behavior looks like. The framework will use several techniques: transforming features, learning patterns over time, contrastive learning, and scoring for anomalies. Features like packet size, how long a connection lasts, times between packets, the type of protocol used, and details about the source and destination will be used to build training examples. An encoder based on Transformer or LSTM models will create hidden representations of the data. Contrastive learning will help the model recognize similar and different traffic patterns. Then, an anomaly score will be calculated based on these representations. The framework will be tested using standard cybersecurity datasets such as CICIDS2017, UNSW-NB15, CIC-Darknet2020, and possibly encrypted traffic datasets. Its performance will be measured using accuracy, precision, recall, F1 score, ROC-AUC, PR-AUC, false-positive rate, and how well it detects new attacks. The goal is to create a scalable and efficient system that can find new and unknown attack patterns without needing labeled data.

Keywords: Self-Supervised Learning; Cybersecurity; Anomaly Detection; Network Intrusion Detection; Contrastive Learning; Deep Learning; Zero-Day Attack; Network Traffic; Transformer; Representation Learning.

Introduction

The fast growth of cloud computing, the Internet of Things (IoT), mobile networks, and connected digital systems has made cybersecurity threats more complex and bigger than before. Cyberattacks like DDoS, botnets, ransom ware, brute-force attacks, phishing, malware, and data theft are getting smarter and harder to detect with traditional security tools. Most traditional systems use signature-based detection, which relies on known attack patterns[1]. These systems are good at finding attacks they've seen before, but they struggle with new, evolving, and zero-day threats. Because of this, there's a big focus on using smarter methods for anomaly detection in cybersecurity. Machine learning (ML) and deep learning (DL) have

shown great promise in detecting anomalies by learning patterns from network traffic, system logs, device activities, and other security data. Supervised learning techniques like Support Vector Machines, Random Forest, Decision Trees, and neural networks work well when we have a lot of labeled training data[1]. However, getting accurate labels for cybersecurity data is tough. It often needs expert knowledge, a lot of time, and manual work. Plus, real-world cybersecurity data is usually very imbalanced, changes a lot, and includes new types of attacks. These issues make it hard for traditional supervised models to adapt to new environments and threats. Self-supervised learning (SSL) is a promising way to solve the labeling problem. Unlike supervised learning, SSL doesn't need each data point to have a manual label. It instead learns from the data itself by creating small tasks, like figuring out which parts of the data are similar. This helps the model understand the features of the data, which can then be used for anomaly detection or other tasks[1]. Recent studies show SSL is becoming more important in cybersecurity, especially for detecting network intrusions and malware. One type of SSL is contrastive learning, where the model is encouraged to group similar views of the same data and push apart unrelated ones. This helps it learn normal behavior patterns without needing attack labels. In cybersecurity, contrastive learning can spot unusual behavior and improve detection of new attacks. Research has also shown its use in encrypted traffic, where patterns in data flow can be studied without looking at the actual content. Another challenge in detecting cyber threats is that network behavior happens over time and depends on context[1-2]. A single packet or flow may not show if something is wrong, but patterns over time might. To handle this, models like Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Transformers can be used with self-supervised learning. These models better understand the timing and relationships between network events than traditional ML methods. Even though there's progress in ML, DL, and SSL for cybersecurity, there are still many challenges. These include high false alarms, imbalanced data, changing attack patterns, poor performance across different datasets, difficulty spotting zero-day attacks, and the lack of transparency in deep learning models. A model trained on one network might not work well on another if the environment or attack types are different[2]. Also, cybersecurity systems need to detect threats quickly while using few resources, especially in high traffic or limited

environments. Because of this, this research proposes a Self-Supervised Learning framework for Cybersecurity Anomaly Detection[3]. The framework learns from mostly unlabeled network data and uses these insights to find unusual or harmful activities. It combines contrastive self-supervised learning with deep temporal models like Transformers or LSTMs, followed by an anomaly-scoring system. The framework will be tested using well-known cybersecurity datasets like CICIDS2017, UNSW-NB15, CSE-CIC-IDS2018, and CIC-Darknet2020.Performance will be measured using precision, recall, F1-score, ROC-AUC, PR-AUC, false-positive rate, and the ability to detect zero-day attacks. The research tries to cut down on the need for people to manually label cybersecurity data, and it also wants to make anomaly detection systems better at finding both familiar and new kinds of threats. The study will use techniques like self-supervised learning to create features, modeling of time-based patterns, and methods to explain how anomalies are detected. These approaches aim to make cybersecurity systems more flexible, easier to scale, more efficient with labels, and stronger in real-world settings where networks are always changing[1-4].

Related Work

Cybersecurity systems for finding unusual activity have changed over time. They used to mainly use signature-based methods, where they looked for known attack patterns. These systems worked well for threats that were already known, but they had trouble finding new or unknown attacks. To solve this, new methods like machine learning, deep learning, and more recently self-supervised learning have been used. These newer methods learn by looking at data to understand what is normal and what is harmful[1].

1. Machine Learning-Based Cybersecurity Anomaly Detection

Early machine learning-based intrusion detection systems often used algorithms like Decision Trees, Random Forest, Support Vector Machines (SVM), k-Nearest Neighbors (KNN), Naïve Bayes, and ensemble methods. These methods can sort network traffic into safe or harmful categories if the right features and labeled data are available. But their success depends a lot on how well the features are chosen and how good the training data is. Also, cybersecurity data sets often have too many examples of normal activity compared to attacks, some extra features that don't help,

and attacks that change over time. These issues can hurt how well the models work in real situations[3-4].

Recent studies have found that the performance of network intrusion detection systems is greatly affected by the quality of the data, how features are selected, how the model is tested, and how varied the attack types are. Because of this, benchmark datasets like UNSW-NB15, CICIDS2017, CSE-CIC-IDS2018, and CIC-Darknet2020 are widely used to test new intrusion detection techniques[3].

2. Deep Learning for Intrusion Detection

Deep learning has increasingly been applied to cybersecurity because neural networks can automatically learn hierarchical representations from high-dimensional network data. Convolutional Neural Networks (CNNs) have been used to capture local relationships among network features, while Recurrent Neural Networks (RNNs), LSTMs, and GRUs have been employed to model sequential network behavior[4].

Autoencoder-based methods are particularly relevant to anomaly detection because they can learn a compact representation of normal traffic and identify observations with high reconstruction errors as potential anomalies. However, conventional autoencoders may learn representations that are not sufficiently discriminative when the normal traffic distribution is complex. Deep-learning-based intrusion detection systems may also require substantial computational resources and can suffer from high false-positive rates when deployed in changing network environments[4].

Recent surveys have highlighted the growing use of CNN, RNN, autoencoder, and Transformer architectures for network intrusion detection and emphasized the need for models capable of handling evolving and previously unseen attacks[1].

3. Self-Supervised Learning

Self-supervised learning has emerged as an important approach for reducing dependence on manually labeled cybersecurity data. Instead of learning directly from predefined attack labels, SSL creates supervisory signals from the structure of the input data. This allows models to learn useful representations from large quantities of unlabeled network traffic. In cybersecurity, this is especially useful because companies keep creating a lot of data from networks, devices, user logins, and systems, but only a small part of this data is correctly labeled. A 2026 survey on self-supervised learning in cybersecurity found that detecting network intrusions and

identifying malware are big areas where this method can be used. It also showed that self-supervised learning can help models learn from data that hasn't been labeled yet. Various strategies for self-supervised learning have been studied, such as learning by rebuilding data, predicting hidden features, predicting future events, grouping similar data, and learning by comparing data. These methods let the model understand how different data points are connected before using that understanding to spot unusual activity or classify data[4-6].

4. Contrastive Learning for Cybersecurity

Contrastive learning has become one of the most promising SSL techniques for cybersecurity anomaly detection. The fundamental principle is to construct positive and negative pairs and train an encoder to produce similar representations for related observations while separating unrelated observations. For network traffic, positive pairs are made by applying different changes to the same network flow or by choosing observations that happen at similar times or show similar behavior. Negative examples come from traffic that is not related[5]. The encoder then learns a way to represent data so that normal and unusual patterns can be told apart more clearly.

Recent studies have shown that self-supervised contrastive learning can be used to detect anomalies in encrypted network traffic. This is important because encryption stops traditional methods of checking the actual content of data. Instead, features like packet sizes, timing, flow length, and other statistical details can be used to learn data representations without looking directly at the encrypted content[5].

5. Transformer-Based Cybersecurity Detection

Transformer architectures have recently attracted attention in cybersecurity because their self-attention mechanism can model relationships between events over relatively long sequences. Unlike conventional recurrent networks, Transformers can process relationships between multiple network events without relying exclusively on sequential recurrence. In intrusion detection, Transformer models can capture temporal and contextual dependencies among network flows, system events, or log sequences[6]. Their combination with self-supervised learning is particularly promising because the Transformer encoder can first learn general representations from unlabeled data and subsequently be adapted for anomaly detection. However, Transformer-

based approaches can have high computational and memory requirements, particularly when processing long network-event sequences. Consequently, model optimization and efficient feature representations remain important research challenges[1].

6. Zero-Day Attack Detection

One big reason people study cybersecurity using SSL is to find new kinds of attacks that haven't been seen before. Normal methods that use labeled data to teach machines how to tell good traffic from bad traffic often fail when they come across a new type of attack because they haven't learned about it before[1]. These methods rely on knowing what types of attacks exist in the training data, so if a new attack shows up, it might be mistaken for normal activity. Another way is to use self-supervised learning, which helps machines understand normal network behavior without needing specific attack labels. If a new attack behaves in a way that's very different from what's normal, it might be spotted as an unusual event even if it wasn't labeled in the training data. But it's still hard to tell if something is a real new attack or just normal activity that's different. Because of this, there's a need for strong ways to score how unusual something is, choosing the right thresholds, looking at how things change over time, and being able to explain why something is flagged as suspicious[6-7].

Research Gap

The literature indicates that significant progress has been made in machine-learning-based, deep-learning-based, and self-supervised cybersecurity detection. However, several research gaps remain[1]:

Area	Existing limitation	Research opportunity
Labeled-data dependency	Supervised models require extensive labeled datasets	Self-supervised pretraining
Zero-day detection	Known attack classes dominate training	Representation-based anomaly detection
Network dynamics	Static models may miss temporal behavior	LSTM/GRU/Transformer
Encrypted traffic	Payload inspection may not be possible	Flow-level SSL
False positives	Legitimate unusual traffic may be misclassified	Context-aware anomaly scoring
Dataset generalization	Models may perform differently across datasets	Cross-dataset evaluation
Class imbalance	Rare attacks are underrepresented	SSL + anomaly detection
Explainability	Deep models can be difficult to interpret	SHAP/attention-based explanations

Concept drift	Network behavior changes over time	Adaptive/continual SSL

So, there's a clear chance to create a single self-supervised system that brings together contrastive learning, time-based analysis, and methods for spotting unusual activity. This system could learn from most of the network traffic that isn't labeled, and at the same time, help find both known, unknown, and brand-new types of attacks[8].

Positioning of the Proposed Research

The proposed research differs from conventional supervised IDS approaches by placing self-supervised representation learning at the core of the detection pipeline. A potential architecture is[7]:

Unlabeled Network Traffic → Preprocessing → Data Augmentation → Contrastive SSL → Transformer/LSTM Encoder → Latent Representation → Anomaly Score → Normal/Anomalous → Explainable Prediction

The study can further evaluate whether SSL pretraining reduces the amount of labeled data required for effective intrusion detection. Experiments using different percentages of labeled data—for example, **1%, 5%, 10%, 25%, 50%, and 100%**—would provide strong evidence of the label-efficiency advantage of the proposed approach[1].

Objective

The objective of this research is to design, develop, and experimentally validate a self-supervised deep learning framework for cybersecurity anomaly detection that can learn meaningful representations from large volumes of unlabeled network traffic and accurately identify normal, anomalous, and previously unseen cyber-attack behaviors with reduced dependence on manually labeled data[6-7].

1. To conduct a comprehensive investigation of existing cybersecurity anomaly detection techniques.

The study will systematically examine traditional signature-based intrusion detection, statistical anomaly detection, machine learning, deep learning, unsupervised learning, and self-supervised learning approaches. Particular attention will be given to their dependence on labeled data, detection accuracy, false-positive rates, computational requirements, ability to detect zero-day attacks, and generalization across different network environments[7].

2. To investigate the characteristics and challenges of cybersecurity network-traffic data.

The research will analyze network-flow characteristics, including packet-level and flow-level statistical features, protocol information, source and destination behavior, flow duration, packet size, inter-arrival time, and temporal patterns. The study will also investigate challenges such as missing values, redundant features, class imbalance, high dimensionality, noisy observations, encrypted traffic, and concept drift[7].

3. To develop an effective preprocessing and feature-representation pipeline for cybersecurity data.

The proposed research will develop procedures for data cleaning, missing-value handling, normalization, categorical-feature encoding, feature selection, dimensionality reduction where appropriate, and removal of redundant information. The objective is to create robust input representations suitable for self-supervised deep-learning models[7].

4. To develop a self-supervised learning mechanism for learning representations from unlabeled network traffic.

The study will design self-supervised pretext tasks that allow the model to generate supervisory information directly from network data. The objective is to enable the model to learn meaningful representations without requiring every network observation to be manually labeled as benign or malicious[6].

5. To develop a contrastive self-supervised learning approach for cybersecurity anomaly detection.

The research will investigate contrastive learning in which multiple augmented views of network observations are generated and used to learn similarities and differences between behavioral patterns. The objective is to produce a latent representation space in which similar network behaviors are closely grouped while substantially different or anomalous behaviors are separated.

6. To incorporate temporal and contextual information into the self-supervised detection framework.

Because cyber-attacks often occur as sequences of related events rather than isolated network flows, the study will investigate LSTM, GRU, or Transformer-based architectures. The objective is to capture temporal dependencies, behavioral sequences, and contextual relationships that

may provide additional evidence for identifying malicious activities[8].

Methodology

The research introduces a new method for detecting unusual or new types of cyber activities in network traffic, using self-supervised learning. This approach helps find what is normal, what is unusual, and what has never been seen before, without depending much on data that has been manually labeled. The framework will be tested using several well-known cybersecurity datasets, including CICIDS2017, UNSW-NB15, CSE-CIC-IDS2018, and CIC-Darknet2020.Before analysis, the data will be cleaned by removing duplicates and invalid entries, filling in missing values, converting text into numbers, scaling the data, and selecting important features[9]. From the network traffic, key characteristics like packet size, flow duration, packet rates, time between packets, used protocols, TCP flags, and traffic patterns in both directions will be extracted. Then, self-supervised learning will be applied to the mostly unlabeled data by creating multiple views of the same traffic through techniques like masking features, adding noise, randomly removing features, and splitting data over time. A contrastive learning method will help the model learn strong representations by bringing similar traffic patterns closer together and pushing apart different ones, using an InfoNCE-based goal. These learned patterns will then be processed with a Transformer or LSTM model to understand the context and order of network behavior[10]. An anomaly score will be generated based on these learned features, and an adaptive threshold will decide whether each activity is normal or abnormal. To test how well the system handles new or unknown attacks, certain attack types will be removed during training and added only during testing. The new method will be compared to existing machine learning, deep learning, and unsupervised models using metrics like accuracy, precision, recall, F1 score, ROC-AUC, PR-AUC, and false-positive rate. Extra tests will look into how efficiently the model uses labels, how well it works across different datasets, how important each part of the system is, and how much computing power it uses. Finally, tools like SHAP, attention analysis, and visualizations using t-SNE or UMAP will be used to make the model's decisions more understandable and to find out which network features most influence the detection of anomalies[10].

Performance Parameters

The performance of the proposed Self-Supervised Learning-based Cybersecurity Anomaly Detection Framework will be evaluated using the following parameters:

Performance Parameter	Formula / Measurement	Purpose
Accuracy	(TP+TN)/(TP+TN+FP+FN)(TP+TN)/(TP+TN+FP+FN)	Measures overall classification correctness
Precision	TP/(TP+FP)TP/(TP+FP)	Measures how many detected anomalies are actually attacks
Recall / Detection Rate	TP/(TP+FN)TP/(TP+FN)	Measures the ability to detect actual attacks
F1-Score	2(PR)/(P+R)2(PR)/(P+R)	Provides a balance between precision and recall
Specificity	TN/(TN+FP)TN/(TN+FP)	Measures the ability to correctly identify normal traffic
False Positive Rate (FPR)	FP/(FP+TN)FP/(FP+TN)	Measures incorrectly detected normal traffic
False Negative Rate (FNR)	FN/(FN+TP)FN/(FN+TP)	Measures missed attacks
ROC-AUC	Area under ROC curve	Measures discrimination between normal and anomalous traffic
PR-AUC	Area under Precision–Recall curve	Particularly useful for imbalanced cybersecurity data
Zero-Day Detection Rate	Detected unseen attacks / Total unseen attacks	Measures capability to identify previously unseen attacks
Inference Time	Milliseconds/sample	Measures real-time detection capability
Throughput	Samples or flows/second	Measures processing capacity
Training Time	Total model training time	Measures computational efficiency
Model Size	Number of parameters / MB	Measures deployment feasibility
Cross-Dataset Performance	Performance across different datasets	Measures generalization capability

For the proposed research, Precision, Recall, F1-score, FPR, PR-AUC, and Zero-Day Detection Rate will be considered particularly important because cybersecurity datasets are often highly imbalanced and the cost of missed attacks or excessive false

alarms can be significant. The proposed model will also be compared with baseline machine-learning and deep-learning models to determine whether self-supervised and contrastive representation learning provides measurable improvements in detection performance[8-9].

Discussion

The new framework is designed to fix some of the problems with traditional cybersecurity tools that detect unusual activity. First, using self-supervised learning means the system doesn't need a lot of data that has been manually labeled. Cybersecurity teams create a lot of network data, but it's costly and hard to label everything properly. By learning from data that hasn't been labeled, the system can scale better and handle more information. Second, contrastive learning helps the model learn better patterns by teaching it to tell the difference between similar and different network activities. This is helpful for finding small changes that regular models might miss. Third, using a Transformer or LSTM lets the system understand how things change over time[10]. This is important for attacks that happen in steps or through multiple events. Fourth, testing the system on zero-day attacks shows how well it works beyond just known threats. A model that only detects attacks it has seen before might not be useful in real situations. Showing it can pick up on new attack types proves it's really good at spotting unusual behavior. Fifth, using multiple datasets helps see if the model can work well in different situations. If it only works on one type of data, it might be too tuned to that specific data and not general enough. Finally, making the model explainable helps cybersecurity experts understand why something was flagged as unusual. This makes it easier to investigate incidents and reduces reliance on unclear predictions.

However, there are some challenges to consider. Real-world network environments may not match the data used for testing. Traffic patterns can change over time, leading to outdated models. Self-supervised learning can be tricky depending on how data is altered, how negative examples are selected, the model's structure, and other settings. Transformer models also need a lot of computing power. These issues should be studied using detailed tests, comparisons, and analysis of performance across different data sources[8].

Conclusion

This study introduces a new cybersecurity framework that uses self-supervised learning

to detect unusual network activity. It aims to improve upon traditional systems that rely heavily on labeled data, which can be expensive and time-consuming to prepare. The framework includes several key parts: data cleaning, learning features from unlabeled data, contrastive learning to understand similarities and differences between network behaviors, time-based analysis to track sequences of network events, scoring to spot unusual activity, testing for new threats, and using AI that can explain its decisions. The main advantage of this framework is its ability to learn useful patterns from mostly unlabeled data, making it less dependent on manual labeling. Contrastive learning helps the model recognize common and different patterns in network behavior, while using Transformer or LSTM models helps understand how network activities change over time. The system then identifies any deviations from normal behavior through a scoring system. The evaluation includes standard measures like accuracy, precision, recall, F1-score, ROC-AUC, and PR-AUC, along with other important factors like how often false alarms happen, how fast the system runs, how well it works with different data sets, and how well it detects new threats.

The study will also test each part of the framework to see how much each part contributes to the overall performance. Overall, this research could help create more adaptable, cost-effective, scalable, and strong systems for detecting cyber threats. Combining self-supervised learning with time-based feature learning offers a good approach for spotting new and changing cyber threats. Future work could expand this framework to support continuous learning, decentralized self-supervised learning, real-time threat detection, resistance to attacks, and deployment in various environments such as businesses, IoT devices, cloud services, and encrypted networks.

REFERENCES

1. Aldweesh, A., Derhab, A., & Emam, A. Z. (2019). Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues. *Knowledge-Based Systems, 189*, 105124.
2. Choi, H., Kim, M., Lee, G., & Kim, W. (2019). Unsupervised learning approach for network intrusion detection system using autoencoders. *The Journal of Supercomputing, 75*(9), 5597–5621.
3. Zhong, Y., Chen, W., Wang, Z., Chen, Y., Wang, K., Li, Y., Yin, X., Shi, X., Yang, J., & Li, K. (2019). HELAD: A novel network anomaly detection model based on heterogeneous ensemble learning. *Computer Networks, 168*, 107049.

4. Lee, J. H., & Park, K. H. (2019). AE-CGAN model based high-performance network intrusion detection system. *Applied Sciences, 9*(20), 4221.
5. Ergen, T., & Kozat, S. S. (2019). Unsupervised anomaly detection with LSTM neural networks. *IEEE Transactions on Neural Networks and Learning Systems, 31*(8), 3127–3141.
7. Hsu, Y. F., He, Z. Y., Tarutani, Y., & Matsuoka, M. (2019). Toward an online network intrusion detection system based on ensemble learning. In *2019 IEEE International Conference on Cloud Computing (CLOUD)* (pp. 174–178). IEEE.
8. Hwang, R. H., Peng, M. C., Huang, C. W., Lin, P. C., & Nguyen, V. L. (2020). An unsupervised deep learning model for early network traffic anomaly detection. *IEEE Access, 8*, 30387–30399.
9. Verkerken, M., D'hooge, L., Wauters, T., Volckaert, B., & De Turck, F. (2020). Unsupervised machine learning techniques for network intrusion detection on modern data. In *2020 Fourth Cyber Security in Networking Conference (CSNet)* (pp. 1–8). IEEE.
10. Vaiyapuri, T., & Binbusayyis, A. (2020). Application of deep autoencoder as a one-class classifier for unsupervised network intrusion detection: A comparative evaluation. *PeerJ Computer Science, 6*, e327.