VOIP Flood Attack Detection with Explainable and Self-Supervised Machine Learning Models

Amisha Kumari Singh , Harsh Vishwakarma
Department of Computer Science and Engineering, Ashoka Institute of Technology and Management, Varanasi

***Abstract*:** Voice over Internet Protocol is one of the most widely used tools in the modern digital communication landscape. However, as with many SIP based VoIP networks, they have become increasingly vulnerable to attacks. This research proposes an ensemble-based machine learning method to detect such attacks with high precision. In our framework, we implemented semi-supervised techniques using autoencoders and advanced NNER-based classifiers like CatBoost,and EBM to add supervised HDBSCAN and Isolation Forests unsupervised models. Extracting features from SIP/RTP packets VoIP systems provided us with a real world dataset. Ensemble models provided up to 99.52% accuracy, vastly exceeding individual models. In addition to the ensemble models perform several predictive and descriptive longitudinal analyses as well as report evaluation metrics. Overall, we confirmed the systems performed well, and multilevel ensemble techniques improve the SIP based VoIP system resilience to flooding attacks.

***Keywords*:** Voice over IP; CatBoost; EBM; Attack; Security; Explainable Boosting Machine, Ensemble Learning; CatBoost; EBM; Register and RTP Flood Attack; Self-Supervised Tabular Learning (SST); Feature Selection.

1. Introduction

Voice over Internet Protocol (VoIP) is a technology that enables voice and multimedia communication over the internet, rather than through traditional telephone networks. It converts analog audio signals into digital data packets, which are then transmitted over IP-based networks. Core protocols such as the **Session Initiation Protocol (SIP)** handle call setup and management, while the **Real-Time Transport Protocol (RTP)** manages the actual transmission of media streams like voice and video.

Key advantages of VoIP are **cost efficiency, high scalability,** and **flexibility** , especially for long-distance and international calls — allowing businesses to expand their communication infrastructure without significant hardware investment. Additionally, VoIP systems support features such as video calls, call forwarding, voicemail, and integration with other digital communication tools — making them highly suitable for both personal and enterprise-level use.

With the increasing shift toward digital and remote communication, VoIP plays a crucial role in modern communication systems — powering everything from corporate call centers and customer support systems to personal video conferencing and unified communication platforms.

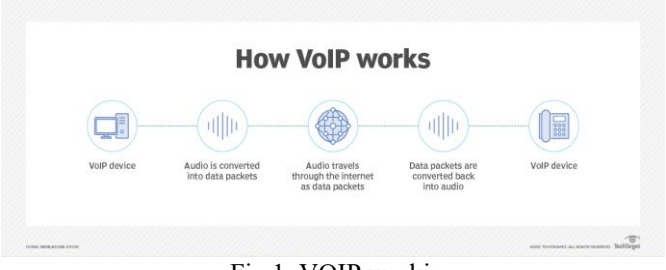

Fig 1. VOIP working

1.1 VoIP Components

Voice over Internet Protocol (VoIP) systems rely on three core components:

Codec(Coder/Decoder):
Converts analog voice signals into digital form, compresses them, and encodes them using voice codecs like G.711, G.729, or G.723.1a. This enables efficient transmission over IP networks.

Packetizer:
Splits the encoded voice into fixed-size packets. Each packet is attached with multiple protocol headers such as RTP (Real-time Transport Protocol), UDP, and IP to support real-time delivery.

Playout Buffer:
Used at the receiver's end to mitigate network jitter (variation in packet arrival times). It holds incoming packets temporarily to ensure smooth audio playback. Late-arriving packets (beyond the playout time) are discarded.

Additionally, UDP is preferred over TCP in VoIP due to its lower latency and lack of acknowledgment overhead, which makes it more suitable for real-time communication.

SignalingProtocols:
Protocols like SIP (Session Initiation Protocol) and H.323 are used to establish and terminate VoIP calls. SIP is designed by IETF for internet applications, while H.323, standardized by ITU-T, integrates smoothly with traditional PSTN networks.

1.2Why VoIP is Vulnerable to Attacks

VoIP systems rely on several open and widely-used protocols such as **SIP** (Session Initiation Protocol) for signaling and **RTP** (Real-Time Transport Protocol) for media transfer. These protocols were originally designed for interoperability and flexibility, not with strong built-in security. As a result, VoIP infrastructure is susceptible to various types of attacks that exploit protocol weaknesses, poor configuration, lack of

encryption, and resource limitations.

Attackers may target:

- **The SIP layer** (e.g., with fake call requests or registration manipulation)
- **The RTP layer** (e.g., with media stream interference)
- **The infrastructure** (e.g., overloading SIP servers, hijacking sessions)

We have seen an increase in malicious attacks on the internet over the past few years as the internet continues to grow and integrate more facets of our everyday life than ever. These attacks are mostly targeted towards communications, payments, and many other aspects [1]. Therefore, the importance for network security professionals to effectively identify these different types of attacks and to prevent them from using various network security techniques runs constant.

Voice over Internet Protocol (VoIP) is technology that uses

connectivity over Internet Protocol (IP) networks to communicate with the system. In addition to traditional phone services including VOIP, it offers voice call flexibility and efficiency like that of the traditional Public Switched Telephone Network (PSTN).
If we compare VoIP to traditional telephony, it has emerged as a standard for voice communication using the Internet and it allows the integration of more communication options and at lower cost compared to traditional telephony.

1.3 DDOS Attack

A **DDoS attack** involves multiple systems (often compromised) working together to flood a target with excessive traffic, aiming to **disrupt availability** by exhausting its bandwidth, processing power, or session-handling capacity. In VoIP networks, DDoS attacks can lead to **call failures**, **service outages**, and **significant revenue and reputation loss**. Distributed Denial of Service (DDoS) attacks are among the most disruptive and malicious threats in modern networked environments. These attacks aim to render a system or service unavailable by overwhelming it with a flood of traffic from multiple distributed sources, often coordinated through botnets. In the context of real-time communication systems like Voice over Internet Protocol (VoIP), the impact of DDoS is especially critical due to the stringent requirements on latency, availability, and continuous media flow. A common form of DDoS is the **flooding attack**, where attackers send a massive volume of protocol-specific messages or packets to overload the target's network bandwidth, processing power, or application state tables.
Within SIP-based VoIP networks, **flooding attacks are highly prevalent**, targeting both signaling and media layers. Two of the most impactful flooding attacks in this domain are the **RTP Flood** and **REGISTER Flood**. **RTP Flood attacks** target the media path by injecting a high volume of fake RTP (Real-time Transport Protocol) packets, thereby consuming network and CPU resources and severely degrading call quality or media delivery

1.4 RTP Flooding Attack

VoIP network requires two level of security mechanism. One for protecting the signaling phases to establishment a media session and, other for protecting the media security. Thus, the media security is strongly coupled with signaling security, because the media sessions are described by the Session Description Protocol (SDP). Many researchers briefly described the SIP flooding2-10. This paper presents the RTP flooding attack; the attacker floods the media packets during media conversation phase as shown in Fig.1. Flooding generates a baggage of valid RTP voice messages (both the header and the payload are filled with random bytes) over UDP to the targeted clients in a dialog phase and makes the server prepare the response messages, with the objective of exhausting bandwidth. This can be achieved by hacking of legitimate user credential information (INVITE). Therefore, the sequence number in succeeding media packets should increase frequently else we must check the RTP packets of legitimate users5. Such consequence of flooding results server overload or crash, unwanted traffic rate, customer service to be delayed or frozen, unsuccessful call completion rate and finally, some innocent server to be overloaded. Thereby, the voice call quality is affected by intermittent voice transfer.

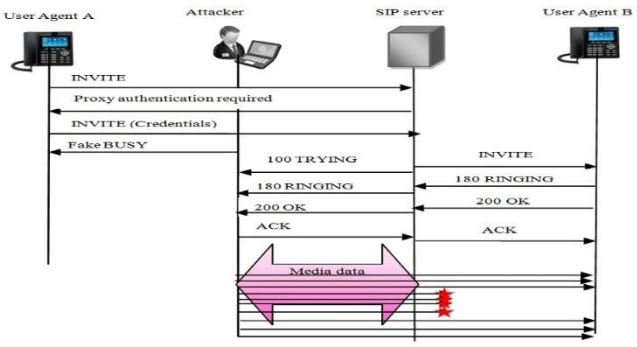

Fig 2 . An example of RTP flooding attack

1.5 Register flooding attack

A REGISTER Flood is a type of signaling-layer flooding attack in which an attacker sends an excessive number of SIP REGISTER requests to a SIP server or registrar. In a typical VoIP environment, a REGISTER request is used by a client (e.g., IP phone or softphone) to inform the SIP server of its current IP address and availability, allowing it to receive incoming calls. However, in a REGISTER flooding scenario, the attacker exploits this process by generating and sending a continuous stream of malformed, spoofed, or fake REGISTER messages.
The goal of a REGISTER Flood is to exhaust server-side resources such as memory, CPU cycles, and registration tables. Each incoming REGISTER request, whether valid or fake, consumes processing time as the server must parse, authenticate, and potentially store the registration. A large volume of such requests can overwhelm the SIP server, causing it to delay or reject legitimate registration attempts, ultimately resulting in service denial for actual users.

2. Literature

The increase in cyberattacks aimed at VoIP infrastructures has led to significant research efforts to develop sophisticated systems for intrusion detection (IDS). VoIP services, especially those that utilize the Session Initiation Protocol (SIP), are known to suffer from several vulnerabilities such as Registration Flood, RTP Flood, SIP fuzzing, and Denial of Service (DoS) attacks. These weaknesses have been the focus of several researchers who have proposed both machine learning and rule-based detection systems to counter them.

2.1 VoIP and SIP Security Threats

The SIP protocol, which forms the backbone of VoIP signaling, is vulnerable to attacks because it lacks proper security measures and is text-based. Abdalla Jama et al. [1] proposed an SVM-based detection model for IP-PBX DoS attacks. Their model effectively SIP-traffic anomaly detection, basing its accuracy on signaling flow metrics. The Sip-traffic anomaly detection model is accurate as it uses signaling flow metrics. But the SVM approach's weaknesses on real-time scalability, and imbalanced datasets, hindered its efficiency.

The work "Dataset of Attacks on a Live Enterprise VoIP Network" published by Iacob and others [2] SIP and RTP attack vectors in an Enterprise VoIP setup are captured in a live enterprise-grade dataset. In addition to showcasing the feasibility of attack emulation in real-time, this paper also offers critical information to develop and evaluate machine learning classifiers. As a result, the **register_flood** and **RTP_flood** attack dataset

served as the baseline for countless machine-learning-based detection systems, including the one presented in this paper.

2.2 Machine Learning in Intrusion Detection

The performance of traditional IDS systems has been improved and enhanced by the implementation of Machine Learning (ML) techniques. These models are capable of learning from traffic and adapting to new attack techniques over time, improving and evolving, unlike traditional IDS systems that rely on hardcoded signatures.

Due to its strong performance and high-rated explainability and interpretability, successful application of the CatBoost algorithm, a gradient boosting algorithm that is optimal for categorical data, has been noted for its success in intrusion detection.

The Explainable Boosting Machine (EBM) is equally known for its explainability and accuracy, making it attractive in security-sensitive domains that require transparency of the decision-making process.

TabNet, an architecture proposed for deep learning on tabular data, has been successful in cases where there are complicated feature interactions among a number of elements.

Through Self-Supervised Tabular Embeddings (SST), feature extraction using autoencoders uncovers relevant patterns within raw feature vectors. This preprocessing step significantly reduces the noise and dimensionality of the feature vectors before using the classifiers such as CatBoost and EBM.

These classifiers have surpassed the benchmarks set by classical methods as they achieve better precision, recall, and F1-score. Their advantage is more pronounced in the case of multi-class and imbalanced datasets.

2.3 Ensemble and Hybrid Models

Recent research highlights the importance of combining different models to create ensemble or hybrid systems. These systems improve resilience and reduce false alarms. Our method follows this trend. Here are some examples:

CatBoost, EBM, and SST Ensemble provides high precision while maintaining clarity and flexibility.

Hybrid setups that include HDBSCAN, which groups data without supervision, and Isolation Forest, which detects anomalies, help identify new types of attacks by mapping normal actions and recognizing differences.

This combined approach aligns with current trends in security studies. Models that learn on their own work alongside guided classifiers to better detect zero-day threats.

2.4 Realistic Network Testing and Dataset Relevance

Many earlier studies in VoIP security used simulated datasets or synthetic packet traces. However, Iacob et al. showed that testing on real-time VoIP networks results in more precise evaluations. By using traffic captured during active SIP sessions under DoS conditions, the dataset reflects true protocol behavior, congestion patterns, and timing of anomalies.

This focus on real data testing matches the recommendations from cybersecurity agencies that support context-aware, scenario-based IDS evaluations instead of static benchmark testing.

Summary

From the reviewed literature, it is clear that while many methods exist for detecting VoIP-based attacks, combining supervised machine learning, unsupervised clustering, and feature-rich representations offers a top solution. This research builds on those findings by implementing, testing, and comparing advanced models like CatBoost, TabNet, EBM, and SST-based hybrids on a live attack dataset.

3. Training Datasets

In this study, we used real-world datasets containing both malicious and benign VoIP traffic to train and evaluate our machine learning models. Our main goal was to detect anomalies and intrusions in SIP-based communication systems.

We gathered two types of VoIP attack datasets from a public research repository:

- Dataset1 –

  This dataset was captured in a simulated IP-PBX environment, containing both benign SIP/RTP traffic and various SIP-based flooding attacks such as RTP Flood, INVITE Flood, and REGISTER Flood. The dataset structure and generation approach were adapted from the methodology described in [7].

- Dataset2 –

  we constructed a second dataset by aggregating SIP-based VoIP traffic from multiple open-source repositories and network captures. This dataset includes both benign and malicious SIP sessions, collected using WiresharkCaptures and publicly available samples hosted on GitHub. The objective was to simulate a diverse and realistic VoIP environment with a wide range of attack behaviors, such as malformed SIP headers, SIP floods, and replay-based attacks.

After merging both datasets , for two attacks RTP and Register flood.

Total Samples: 189,070 SIP traffic records (combining both attack and normal traffic)

Final Selected Features: We applied various feature selection techniques to narrow down the most important attributes for accurate intrusion classification.

We extracted features from raw SIP and RTP packet flows using tools like Wireshark

4. Proposed Method

In this project, we propose a hybrid approach that combines supervised and unsupervised machine learning to detect specific network intrusions targeting VoIP. Our method analyzes SIP (Session Initiation Protocol) and RTP (Real-Time Transport Protocol) traffic to find unusual communication patterns and DoS (Denial-of-Service) attacks aimed at IP-PBX systems.

The proposed method has the following key stages:

4.1 Dataset Collection and Integration

We used two real VoIP attack datasets:

- Register Flood Attack
- RTP Flood Attack

These datasets were merged and processed to create a combined dataset with 189,070 SIP traffic records. Each record represents a single SIP session or RTP stream and

includes several protocol-level attributes.

4.2 Feature Engineering and Selection

From the raw traffic, we extracted protocol-level behavioral features such as shown in figure.

To reduce noise and dimensionality, we applied a combination of **feature selection ML techniques**, including:

- **Variance Thresholding**
- **Mutual Information**
- **ANOVA F-test**
- **Recursive Feature Elimination (RFE)**
- **L1-Based Feature Selection (Logistic Regression)**
- **Forward Feature Selection (SFS)**
- **Random Forest Feature**

4.3 Feature Selection Techniques

Feature selection is an important step in creating an effective and accurate intrusion detection system. In our project, we first extracted over 45 raw SIP/RTP traffic features from the combined dataset. However, not all features help in telling apart normal and malicious traffic. Redundant and irrelevant features can add complexity and reduce the performance of machine learning models.

To solve this, we used seven different feature selection techniques to find the most informative and relevant features for training. These techniques assisted in reducing dimensionality and improved the generalization ability of our classifiers.

Below is a summary of the selection algorithms used:

4.3.1 Variance Threshold

- Removes features with low variance across samples.
- Assumes that features with near-zero variance are not useful for distinguishing between classes.

Helps in getting rid of constant or nearly constant features.

```
# 4. Variance Threshold
vt = VarianceThreshold(threshold=0.1)
vt.fit(X_train)
vt_features = [feature_cols[i] for i in vt.get_support(indices=True)]
for f in vt_features:
    feature_votes[f] += 1
```

4.3.2 Mutual Information

- Selects top K features based on mutual information score between each feature and the target label.
- Measures dependency between variables.

```
# 1. Mutual Information (non-deterministic)
mi = SelectKBest(score_func=mutual_info_classif, k='all').fit(X_train, y_train)
mi_scores = dict(zip(feature_cols, mi.scores_))
top_mi = sorted(mi_scores.items(), key=lambda x: x[1], reverse=True)[:3]
for f, _ in top_mi:
    feature_votes[f] += 1
```

4.3.3 ANOVA F-test

- Selects features based on F-values, which measure the linear relationship between input features and the output class.
- This approach is helpful for features that are normally distributed.

```
# 2. ANOVA F-test (deterministic)
anova = SelectKBest(score_func=f_classif, k='all').fit(X_train, y_train)
anova_scores = dict(zip(feature_cols, anova.scores_))
top_anova = sorted(anova_scores.items(), key=lambda x: x[1], reverse=True)[:3]
for f, _ in top_anova:
    feature_votes[f] += 1
```

4.3.4 Recursive Feature Elimination (RFE)

- This method removes the least important features using a base estimator, such as Logistic Regression or Random Forest.
- It selects features by repeatedly considering smaller and smaller sets.

```
# 5. RFE
rfe_model = LogisticRegression(random_state=42)
rfe = RFE(estimator=rfe_model, n_features_to_select=3)
rfe.fit(X_train, y_train)
rfe_features = [f for f, s in zip(feature_cols, rfe.support_) if s]
for f in rfe_features:
    feature_votes[f] += 1
```

4.3.5 L1-Based Feature Selection (Logistic Regression)

- Uses L1 penalty in logistic regression to reduce irrelevant feature coefficients to zero.
- Keeps only the most important features.

```
# 3. L1 Logistic Regression
l1_model = LogisticRegression(penalty='l1', solver='liblinear', random_state=42)
l1_model.fit(X_train, y_train)
for f, c in zip(feature_cols, l1_model.coef_[0]):
    if c != 0:
        feature_votes[f] += 1
```

4.3.6 Forward Feature Selection (SFS)

- Sequential Forward Selection adds one feature at a time that boosts model performance the most, using cross-validation to check accuracy.

```
# 6. Forward Selection
fs_model = LogisticRegression(random_state=42)
sfs = SFS(fs_model, k_features=3, forward=True, floating=False, scoring='accuracy', cv=5)
sfs.fit(X_train, y_train)
forward_features = list(sfs.k_feature_names_)
for f in forward_features:
    feature_votes[f] += 1
```

4.3.7 Random Forest Feature

- Random Forests give a feature importance score. This score shows how often a feature helps to split decision nodes in all the trees.

```
# 7. Random Forest Feature Importances
rf = RandomForestClassifier(random_state=42)
rf.fit(X_train, y_train)
rf_scores = dict(zip(feature_cols, rf.feature_importances_))
top_rf = sorted(rf_scores.items(), key=lambda x: x[1], reverse=True)[:3]
for f, _ in top_rf:
    feature_votes[f] += 1
```

This resulted in a **final subset of highly discriminative features** that improved model accuracy and training speed.

List of features

```
"call_id": call_id,
"invite_count": invites,
"register_count": registers,
"response_2xx_count": responses_2xx,
"response_4xx_count": responses_4xx,
"sip_method_diversity": len(set(methods)),
"sip_success_rate": round(responses_2xx / invites, 2) if invites else 0.0,
"sip_session_duration": round(session_duration, 3),
"avg_inter_arrival_time": round(avg_iat, 3),
"user_agent_entropy": round(user_agent_entropy, 3),
"branch_reuse_rate": round(branch_reuse_rate, 3),
"ack_missing_ratio": round(ack_missing_ratio, 3),
"rtcp_presence": int(rtcp_present),
"unusual_port_flag": int(unusual_ports),
"sip_uri_suspicion_flag": int(suspicious_uri),
"cseq_gap_anomaly": int(cseq_gap),
"max_forwards_low_flag": int(max_forwards_low),
"max_forwards_missing_flag": int(max_forwards_missing),
"src_ip": src_ips[0] if src_ips else "",
"dst_ip": dst_ips[0] if dst_ips else "",
"from_uri": from_uris[0] if from_uris else "",
"to_uri": to_uris[0] if to_uris else "",
"packet_count": packet_count,
"class": attack_name
```

4.4 Model Training and Evaluation

We trained and evaluated five machine learning models using the features we selected:

Model	Type	Purpose
CatBoost	Supervised	High-performance classification
Explainable Boosting Machine (EBM)	Supervised	Transparent and interpretable model
Logistic Regression	Supervised	Lightweight, interpretable baseline
SST + Classifier	Self-Supervised + Supervised	Embedding-based learning for better feature representation
HDBSCAN + Isolation Forest	Unsupervised	Anomaly detection for unknown or zero-day attacks

4.5 Model Interpretation and Anomaly Detection

- EBM and Logistic Regression models provided insights into the most influential features in intrusion classification.
- The HDBSCAN and Isolation Forest model helped detect unknown attacks or unusual traffic patterns not present during training.
- The SST embeddings improved performance by learning deeper representations of tabular SIP traffic features.

5. Machine Learning

The Machine learning is a process for extracting knowledge from vast amounts of data. Machine learning models involve the application of a set of rules, methods, or complex "transfer functions" that can be applied to discover or identify similar trends.

In the past few years, machine learning methods have been used to analyze patterns in historical time-series data. One of the solutions to the problem of timely attack detection is to develop a classifier based on machine learning, that would as certain if the incoming traffic has been under threat.

There are three well-known learning types in machine learning as follows: -

• Supervised
• Unsupervised
• Reinforcement learning

Surveyed learning-based IDS strategies use the labeled training data to detect intrusions. This paper is based on Supervised learning.

The supervised approach to learning usually consists of two stages, namely training and testing. Relevant features and classes are defined during the training phase, and the algorithm will then learn from these data samples. Each record in supervised learning IDS is a pair containing a network or host data source and an associated output value, namely intrusion or regular output.

The supervised learning technique is then used to train the classifier using the training data for selected features to learn the inherent relationship between the input data and the labeled output value. During the test phase, the trained model is used to classify the unknown data into an intrusion or a regular class.

The resulting classifier will then become a model that predicts the class to which input data may belong, given the set of values of the feature. Figure 1 demonstrates a general method for applying the techniques for classification.

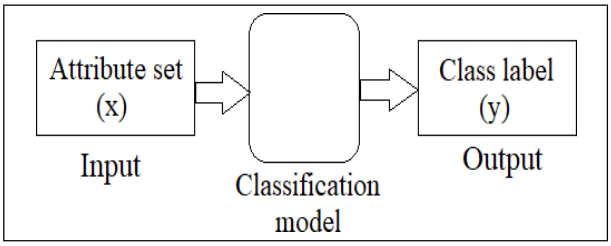

Figure 1. General classification technique

The performance of the classifier in its ability to predict the correct class is measured in terms of several metrics as discussed in Section 4.

There are several classification methods, such as decision

trees, support vector machines and Naïve Bayes. Each technique uses a learning method to create a classification model.
However, the training data should not only be treated with an acceptable classification method but should also properly classify the class of records that it has never seen before. It is a key task of the learning algorithm to construct classification models with a reliable generalization capability.

4.4 Models Used in Our Project

For this project, we looked at five machine learning models to detect VoIP-specific network intrusions. We chose each model based on how well it performed on structured tabular data, its ability to generalize, and how easy it is to understand.

A. CatBoost Classifier

CatBoost is a gradient boosting decision tree algorithm developed by Yandex. It is particularly effective for categorical and tabular data. CatBoost was chosen for its:

- High accuracy
- Robustness to overfitting
- Support for categorical features without preprocessing

B. Explainable Boosting Machine (EBM)

EBM is an interpretable model based on Generalized Additive Models (GAMs). It provides:

- Transparency in feature contribution
- Competitive accuracy compared to black-box models
- Easy explainability for cybersecurity analysts

This model is useful in sensitive environments where understanding model predictions is as important as accuracy.

C. Logistic Regression

Logistic Regression is a baseline linear classifier that predicts the probability of a class label. Despite its simplicity, it is useful due to:

- Fast training and prediction
- Interpretability
- Benchmarking for other complex models

It helps identify how individual features contribute to the classification decision.

D. Self-Supervised Tabular (SST) Embeddings + Classifier

SST is a self-supervised representation learning method designed specifically for tabular data. We trained an SST model to generate embeddings of VoIP traffic and then fed those embeddings into a classifier. Benefits include:

- Rich feature representation
- Improved classification accuracy
- Robust performance on small datasets

E. Unsupervised HDBSCAN + Isolation Forest

To incorporate anomaly detection, we used a hybrid unsupervised approach:

- HDBSCAN (Hierarchical Density-Based Spatial Clustering) to identify dense patterns
- Isolation Forest to isolate anomalies in low-density regions

This ensemble helps detect zero-day or unknown attacks that do not appear in the training set.

5.Experiments and Results

5.1 Introduction to the Experimental Setup

To evaluate the effectiveness of our proposed intrusion detection system for VoIP, we conducted a series of experiments using two datasets: Dataset A (7) and Dataset B (briefly described). We used multiple machine learning and ensemble models for classification and detection, including CatBoost, EBM, and Self-Supervised Tabular Models (SST). All experiments were implemented using Python 3.10 on a system with AMD Ryzen 5 5600H processor, 16 GB RAM, and a 6 GB dedicated graphics card, running a 64-bit Windows operating system.

5.2 Selected feature extraction

In order to reduce computational overhead and enhance model interpretability, feature selection was carried out using a voting-based strategy across multiple selection algorithms.
Among all the extracted SIP session features **max_forwards_missing_flag** and **sip_session_duration** consistently received the highest vote counts in both datasets. These two features were thus selected as the most informative and were used for training and evaluating all models.
This approach ensured that the models focused on the most relevant behavioral indicators without being affected by noisy or redundant attributes.

5.3 Models Used

To evaluate the effectiveness of various machine learning approaches in detecting VoIP-based attacks, we employed a combination of individual and ensemble models.

- CatBoost
- Explainable Boosting Machine (EBM)
- SST + CatBoost
- SST + EBM
- SST + CatBoost + EBM
- SST + CatBoost + HDBSCAN + Isolation Forest

5.4 Evaluation Metrics

Each model was evaluated using a **5-fold cross-validation** approach to ensure robustness and generalizability of results. The following performance metrics were used:

- Accuracy
- Precision
- Recall
- F1-Score
- Confusion Matrix

In addition to classification performance, **computation time** was recorded for each model to evaluate **computational efficiency**, which is particularly important for real-time VoIP intrusion detection scenarios.

5.5 Results and Discussion

We evaluated the selected models on two datasets using a 5-fold cross-validation strategy. Each model was assessed using accuracy, precision, recall, and F1-score. The results for both

datasets are presented below.

Results on Dataset A

Model	Accuracy	Precision	Recall	F1 Score
CatBoost	0.9984	0.9987	0.9988	0.9985
EBM	0.9983	0.9985	0.9988	0.9986
SST+CatBoost	0.9984	0.9988	0.9987	0.9987
SST+EBM	0.9983	0.9988	0.9988	0.9987
SST+CatBoost + EBM	0.9983	0.998	0.9988	0.9987
SST+CatBoost + HDBSCAN + IF	0.9983	0.9988	0.9988	0.9987

All models performed exceptionally well on Dataset 1 with F1-scores above 0.998. While ensemble approaches like SST+CatBoost and SST+CatBoost+HDBSCAN+IF offered marginal improvements in precision or recall, the CatBoost model alone achieved comparable performance with a simpler structure, making it a strong candidate for further evaluation.

Results on Dataset B

Model	Accuracy	Precision	Recall	F1 Score
CatBoost	0.9992	0.9991	0.9997	0.9993
EBM	0.9992	0.9991	0.9998	0.9992
SST+CatBoost	0.9992	0.9991	0.9995	0.9992
SST+CatBoost	0.9990	0.9993	0.9998	0.9990
SST+CatBoost + EBM	0.9991	0.9995	0.9998	0.9992
SST+CatBoost + HDBSCAN + IF	0.9990	0.9992	0.9996	0.9991

Similar trends were observed on Dataset 2, where CatBoost achieved the highest F1-score (0.9993) among all models. The marginal gains offered by ensemble and hybrid models did not justify the added complexity. This consistency across datasets reinforces the robustness and generalizability of the CatBoost model.

Although complex ensemble models such as SST+CatBoost+EBM or SST+CatBoost+HDBSCAN+IF demonstrated slightly better precision or F1 scores (by less than 0.02%), these improvements were marginal and came at the cost of increased model complexity and computation. Following the principle of Occam's Razor in machine learning — *"Among competing models that perform similarly, the simplest one should be preferred"* — we selected the CatBoost model. It consistently delivered high performance across both datasets while maintaining simplicity and efficiency. Therefore, **CatBoost was chosen as the final model** for deployment and further interpretation.

6.Conclusions

This study presents a machine learning-based approach for detecting and classifying VoIP-based SIP DoS attacks in real-time environments. The research explored common SIP-layer attack types such as INVITE floods, REGISTER floods, and RTP-based floods, and analysed their behavioural patterns across multiple SIP sessions.

A new feature extraction pipeline was designed using SIP protocol like Max-Forwards-missing-flags and sip-session-duration metadata such as call duration, offering critical insight into SIP message behaviour. Multiple machine learning models were trained and evaluated, including individual classifiers (e.g., CatBoost, Explainable Boosting Machine) as well as hybrid ensemble combinations with HDBSCAN + Isolation Forest and self-supervised approaches.

The models were evaluated on two datasets. A 5-fold cross-validation technique was used for robust evaluation. Results were measured in terms of accuracy, precision, recall, F1-score, confusion matrix, and computation time.

Although hybrid and ensemble models such as SST+CatBoost+EBM offered marginal improvements in specific metrics, the standalone CatBoost model demonstrated consistently high performance across both datasets (F1: 0.9985 and 0.9993 respectively). Given its simplicity, efficiency, and comparable or superior results, we selected CatBoost as the final model for deployment.

The results indicate that domain-specific features and real-time behavioural profiling can significantly improve VoIP attack detection. To conclude, the proposed detection pipeline and multi-model evaluation framework provide an effective mechanism for proactive SIP DoS detection. Future directions include adapting the model to handle multi-step and evolving attack patterns, as well as testing its integration within live IP-PBX environments. This research also underscores the pressing need for current, annotated VoIP datasets to support the development of robust, ML-based network intrusion detection systems suitable for deployment in real-world networks.

7. References

[1] Jama, A. M., Khalifa, O. O., & Subramaniam, N. K. (2021). "Novel Approach for IP-PBX Denial of Service Intrusion Detection Using Support Vector Machine Algorithm.

[2] W. Nazih, Y. Hifny, W. S. Elkilani, and T. Abdelkader, "Fast Detection of Distributed Denial of Service Attacks in VoIP Networks Using Convolutional Neural Networks," *International Journal of Intelligent Computing and Information Sciences (IJICIS)*, vol. 20, no. 2, pp. 125–138, 2021, doi: 10.21608/ijicis.2021.51555.1046

[3] M. K. Ranganathan and L. Kilmartin, "Performance analysis of secure session initiation protocol based VoIP networks," *Computer Communications*, vol. 25, no. 12, pp. 1139–1150, Jul. 2002, doi: 10.1016/S0140-3664(02)00146-9

[4] .Y. Bouzida, C. Mangin, A framework for detecting anomalies in VoIP networks", in: ARES 2008 - 3rd Int. Conf. Availability, Secur. Reliab. Proc, 2008,pp.204.211,https://doi.org/10.1109/ARES.2008.205.

[5] Chauhan, N. Mahajan, H. Kumar, and S. Kaushal, "Analysis of DDoS Attacks in Heterogeneous VoIP Networks: A Survey," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 6S3, pp. –, Apr. 2019. ISSN: 2278-3075.

[6] Abdalla Jama et al., *"Novel SVM Approach for IP-PBX DoS Detection"*, 2021.

[7] D. Iacob et al., *"Dataset of attacks on a live enterprise VoIP network for machine learning-based intrusion detection and prevention systems"*, Computer Networks, 2021.

[8] T. G. Rahangdale, P. A. Tijare, and S. N. Sawalkar, "An

Overview on Security Analysis of Session Initiation Protocol in VoIP Network," *International Journal of Research in Advent Technology*, vol. 2, no. 4, pp. 190–194, Apr. 2014. E-ISSN: 2321-9637.

[9] C. Choti, N. Hnoohom, S. Tritilanunt, and S. Yuenyong, "Prediction of Intrusion Detection in Voice over Internet Protocol System using Machine Learning," in *Proc. 9th Int. Conf. Comput. Commun. Manag. (ICCCM)*, 2021,https://doi.org/10.1145/3479162.3479185

[10] Wireshark Foundation, "Wireshark: Network Protocol Analyzer," [Online]. Available: https://www.wireshark.org/

[11] Python Software Foundation, "Python Programming Language," [Online]. Available: https://www.python.org/