International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(1s) 2026 pp. 1-24
https://doi.org/10.70917/ijcisim-2026-2015
Article

Integrated Machine Learning and Deep Learning Framework Using Time-Interval Features for Binary and Multiclass Ethereum Threat Detection

Chaitali Deshmukh[1]*,[2], Varsha H. Patil[3]*

[1] Matoshri College of Engineering & Research Centre, Nashik, Maharashtra, India.
[2] K. K. Wagh Institute of Engineering Education and Research, Nashik, Maharashtra, India. chaitalideshmukh@yahoo.com
[3] Matoshri College of Engineering & Research Centre, Nashik, Maharashtra, India. varsha.patil@matoshri.edu.in

Abstract: - Ethereum, the most popular programmable blockchain platform in the world, is now the main object of an ever-growing variety of many malicious organizations, such as phishing attacks, Ponzi scams, cryptocurrency mixers, and fraudulent smart contracts. The ever-growing complexity and scale of on-chain malicious activities require smart and automated detection systems that can simultaneously perform binary threat detection, i.e. identify malicious and benign accounts, and fine-grained multi-category threat detection, i.e. different attack types. The paper describes an Integrated Machine Learning Framework using which the time-interval behavioural features based on the EtherShield dataset are systematically used to provide comprehensive detection of two types of Ethereum threats. The single architecture proposal will avoid the use of different pipeline architectures because it can support binary and multiclass classification paradigms of classification in a single system. Time-interval characteristics are learned over four time windows, which are 1-day, 3-day, 7-day, and 30-day aggregations, to learn time-varying dynamics of Ethereum address behavior. An envelope of monitored machine learning frameworks, comprising of the Random Forest, Gradient Boosting, XGBoost, and Support Vector machine, and deep learning designs, including Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), and Bidirectional LSTM (BiLSTM) are carefully assessed. New hybrid models are suggested: An Isolation Forest enhanced with random Forest model (anomaly-enhanced classification), and a CNN-BiLSTM fusion model (binary sequential analysis). Isolation Forest anomaly scores provide a strengthening of the feature space by the use of unsupervised behavioural signals. It has been experimentally demonstrated that the hybrid CNN+BiLSTM model with binary classification yields 98.89% accuracy whereas the ISO+RF hybrid with multiclass threat detection yields 96.12 % accuracy providing a state of the art standard with blockchain threat intelligence systems.

Keywords: Ethereum Security, Time-Interval Analysis, Machine Learning, Binary Classification, Multiclass Classification, Hybrid Models, Blockchain Analytics

1. Introduction

One of the most disruptive breakthroughs of the digital age has been the blockchain technology, which has created decentralized, unchangeable, and transparent systems of recording financial transactions, as well as carrying out smart contracts. Ethernet, released in 2015 by Vitalik Buterin, built upon the original paradigm of Bitcoin, providing programmable blockchain execution in its Turing-complete scripting language, and enabling decentralized finance (DeFi), non-fungible tokens (NFTs), and multilateral smart contract implementation protocols [1]. Ethereium has gained billions of dollars of digital resources and thousands of decentralized applications that have spawned great economic activity, but has also become a rich target to adversarial exploitation. The anonymous character of blockchain addresses and the irreversibility of the performance of transactions makes the pseudonymous nature of the addresses particularly challenging to threat detect, making conventional cybersecurity strategies have a low probability of success in such scenarios [2].

The circulation of evil practices on the Ethereum platform has escalated to worrying percentages. Hundreds of millions of dollars have been lost every year by phishing attacks, in which attackers impersonate legitimate parties to steal credentials or private keys [3]. Ponzi schemes masquerade as legitimate investment agreements and take the advantage of smart contract logic to defraud the participants in an organized way until they fall [4]. Cryptocurrency mixers, which aim at obstructing the transaction history and launder illegal money, have been developed to be more advanced money laundering devices on blockchain networks [5]. Also, fake tokens and fake initial coin offerings (ICOs) are still on the rise, taking advantage of the permission less character of the Ethereum token creation system. The various typologies of attacks have different behavioural patterns in their transaction history, time-interval distribution and gas consumption signatures, which opens a possibility of machine learning based behavioural analysis [6].

The current methods of detection in blockchain security are intense on rule-based heuristics, offline address blacklists, or address-specific single-model classifiers educated on small sets of features [7]. These methods have serious drawbacks: they do not extend to new types of attacks, are not able to distinguish between different sets of threats occurring at the same time, and do not take into account the rich temporal behavioral information present in sequences of transactions. Moreover, the majority of past studies focus on binary detection, that is, on marking an address as malicious or benign, or multiclass classification, but they seldom combine both into one system. Such fragmentation will require redundant infrastructure, cause additional operational overheads, and can create inconsistency in threat intelligence pipelines used in operational monitoring systems [8]. Behavioral analysis Time-interval the behavioral analysis has become an effective paradigm to the dynamic nature of the malicious Ethereum account activity.

The proposed framework is effective because it takes a pivotal role in picking short-term anomalous spikes and long-term structural behavioral trends by aggregating features across a 1-day, 3-day, 7-day, and 30-day window. This is achieved by the temporal granularity which offers discriminative power which purely statical feature approaches lack by definition. The rationale behind the existence of a unified binary and multiclass framework is necessitated by practical deployment needs: in the real world, blockchain monitoring systems need a unified paradigm to both indicate suspicious accounts on scale and be able to offer fine-grained threat category indicators to trigger response through investigation. The proposed framework addresses this gap by using a common preprocessing and feature extraction pipeline, followed by parallel hybrid CNN–BiLSTM branches that generate both binary detection and multiclass classification outputs. The CNN layers extract discriminative local patterns from the input features, while the BiLSTM layers capture bidirectional sequential dependencies, enabling effective attack detection and classification. The framework evaluates multiple models under identical experimental configurations for both binary and multiclass classification tasks, thereby enabling comprehensive benchmarking and cross-paradigm performance analysis.

The main contributions of the work are the following:

- *Integrated Dual-Mode Framework:* It proposes a unified architecture which concurrently performs binary (malicious vs. benign) and multiclass (Coin Stolen, High Risk, Honeypot, Phishing, Ponzi, Non-Malicious) Ethereum threat identification in a single coherent pipeline, and avoids repetitive pre-processing and the ability to share feature representations.
- *Time-Interval Feature Engineering:* It constructs a time-interval feature engineering model that produces behavioral features in four time windows (1-day, 3-day, 7-day, 30-day) to identify both short-term and long-term malicious trends of Ethereum transaction histories.
- *Model benchmarking:* A set of 9 model architectures (traditional machine learning RF, GB, XGBoost, SVM), deep learning CNN, LSTM, BiLSTM and new hybrid combinations are assessed and compared on binary and multiclass detection of EtherShield benchmark data.
- *New Hybrid Learning Architectures:* Two hybrid approaches are proposed: (i) an Isolation Forest–Random Forest (IF–RF) feature selection framework, where Isolation Forest is utilized to identify anomalous data characteristics and Random Forest computes feature importance scores to rank and select the most informative features, thereby reducing dimensionality and enhancing model effectiveness; and (ii) a CNN–BiLSTM hybrid model that combines convolutional feature extraction with bidirectional sequential learning to perform both binary and multiclass classification of Ethereum account activities.

The remainder of this paper is organized as follows. Section II presents the literature review, while Section III discusses related work. Section IV introduces the proposed framework. Section V describes the EtherShield dataset and preprocessing steps. Section VI details the methodology, and Section VII presents the results and discussion. Finally, Section VIII concludes the paper and outlines future research directions.

2. Literature Review

The list of the detected malicious activity on the blockchain networks has become a research focus in the last ten years, and the increase in Ethereum-based fraud and criminal activity has led to the demand of more automated and scalable detection solutions. The initial research in this area has utilized mostly graph-theoretic and heuristic-driven designs but the massive availability of transaction records has led to a re-orientation towards machine-learning designs. With the capabilities of transaction-level transaction-related aspects, including gas usage patterns, frequency of transactions, and balance movement, scholars have shown the possibility of supervised classification to classify Ethereum threats [9]. Gradient Boosting classifiers, whereas, and Random Forest were first movers in the field, providing excellent performance when used on tabular characterizations of blockchain features with fairly low computing costs and good interpretability via feature importance scores [10].

The importance of temporal and behavioral feature engineering on the enhancement of detection accuracy has been gradually acknowledged. Research has shown that the application of static aggregated features is helpful but cannot reproduce the dynamic behavioral development of malicious Ethereum accounts [11]. The feature extraction based on time, which considers the inter-transaction time variation, bursts of transaction frequency and rolling temporal window statistics, significantly enhances the discriminative power, as they encode the behavioural patterns that vary over both short, medium, and long-term. Studies of anomaly-based blockchain analysis have also underscored the fact that abnormal timing distribution of transactions - typical of mixer protocols and coordinated scam deployments - can be exploited as highly discriminatory behavioral signatures that can be identified by purely value-based properties [12].

Ethereum security literature has treated these two paradigms separately in the field of binary versus multiclass classification. Binary classification studies are coarse, malicious versus benign discrimination, and high-accuracy at simpler model architectures, whereas multiclass studies have extra issues of thin class boundaries and the catastrophic class ratio of threat types, including Coin Stolen, Honeypot, Phishing, Ponzi, Non-Malicious [13]. Both paradigms of classification have been demonstrated to be generalized more successfully by ensemble techniques (especially XGBoost with regularization) because feature sparsity and overfitting are addressed implicitly in ensemble models [14].

The deep learning methods have brought forth novel features of sequential analysis of Ethereum transactions. Convolutional Neural Networks on one dimensional sequences of transaction features have also been shown to be effective at extracting local patterns, including periodic and bursty transaction patterns, [15]. LSTM and Bidirectional LSTM networks have proved to be especially useful in modeling long temporal interactions in transactions histories, with its gating mechanisms permitting the selective retention of useful historical behavioral cues. Face-to-face and reversible sequencing in the BiLSTM architecture has demonstrated better classification performance in challenging multi-class threat situations where future context is able to identify discriminative information about past behavior intention [16].

Combination of unsupervised and supervised learning has also been a particularly promising line of research. A successful application of Isolation Forest, a simple and unsupervised anomaly detector, which relies on recursive random partitioning, has been used to detect statistical outliers of Ethereum transaction behavior, especially when specific types of attack are rare in training data [17]. The incorporation of Isolation Forest anomaly scores as extra features by supervised Random Forest classifiers has been demonstrated to add behavioral cues, which complement their behavior and enhances the detection of previously unseen attack patterns. Even Graph Neural Network has become a useful method of modeling the structural relational trends in Ethereum transaction graphs, but is computationally complex to scale up in practice [18].

In spite of this development, there are still major gaps in research in the current body of work. To begin with, the majority of the research focuses on binary or multiclass detection as independent problems without the integrated frameworks, which would present the feature representations of both paradigms to obtain consistent threat intelligence. Second, the multi-window time-interval features of behavioral dynamics of 1-day, 3 -day, 7-day, and 30-day horizons have not been systematically utilized with hybrid model architectures. Third, the EtherShield, the dataset offering detailed time-interval annotations based on various types of threats has not been used fully in the

literature. Fourthly, the fusion of CNN and BiLSTM into a fusion architecture to detect binary Ethereum threats is a new direction that has not been explored. All these gaps drive the holistic framework suggested in this study.

TABLE I: Summary of Related Work in Ethereum Threat Detection

Ref.	Dataset	Models Used	Features	Task Type	Result Analysis (Accuracy)	Key Limitation
[9]	Ethereum Tx Data	RF, NB	Tx count, Gas, Balance	Binary	95.40%	Static features only
[10]	Phishing Dataset	RF, GB, XGBoost	Graph + Tx statistics	Binary	96.87%	No temporal analysis
[11]	EtherScan Labels	LSTM, GRU	Time-series Tx sequences	Binary	94.23%	Limited class diversity
[12]	Mixer Tx Logs	Isolation Forest	Inter-tx time intervals	Anomaly	91.56%	Unsupervised only
[13]	Labeled Smart Contracts	SVM, RF	Bytecode + Tx stats	Multiclass	89.34%	5-class imbalance
[14]	XBlock-ETH	XGBoost, LightGBM	Address behavior stats	Binary	97.12%	Single time window
[15]	Ethereum Fraud Dataset	1D-CNN	Temporal Tx features	Binary	96.45%	No sequential context
[16]	Ponzi Contract Dataset	BiLSTM	Sequential bytecode + logs	Binary	95.78%	Contract-only scope
[17]	Multi-label ETH data	ISO Forest + RF	Anomaly scores + Tx stats	Binary	97.34%	No multiclass support
[18]	ETH Transaction Graph	GNN, GraphSAGE	Graph structural features	Multiclass	96.01%	High computational cost
[19]	Darknet Market Data	CNN + LSTM	On-chain behavioral features	Multiclass	93.67%	Narrow threat scope
[20]	Smart contract bytecode/source datasets	Deep Learning-based framework (DL4SC); GNN variants	Bytecode patterns, control flow graphs, vulnerability signatures	Vulnerability detection in smart contracts	Not specified	Focused solely on smart contracts; generalizability to other blockchain platforms not addressed
[21]	Blockchain Fraud DB	RF, SVM, DL	Temporal + structural features	Multiclass	94.82%	Missing temporal windows
[22]	EtherShield Dataset	XGBoost, RF	Time-interval features	Both	97.89%	No hybrid integration

[23]	Wireless Sensor Network (WSN) simulation datasets; -	Discrete mathematics-based cryptographic algorithms	Key exchange efficiency, computational overhead, entropy metrics	Cryptographic algorithm design for WSN	Not specified	Adversarial robustness not fully validated
[24]	CICIDS	Random Forest, LSTM,	Network traffic features,	Real-time threat detection and mitigation in IoT	Not specified	Real-time constraint evaluation limited;

Abbreviation:

ML = Machine Learning; DL = Deep Learning; RF = Random Forest; XGB = XGBoost; GNN = Graph Neural Network; SVM = Support Vector Machine; LSTM = Long Short-Term Memory; ISO = Isolation Forest; Acc = Accuracy.

3. Related Work

A. Ethereum Transaction Behavior Modeling

Ethereum transaction behavior can be represented as a time-ordered sequence of interactions between pseudonymous blockchain addresses. For a given Ethereum address aaa, the transaction history is modeled as:

$$T^a = [(t_1, v_1), (t_2, v_2), \dots, (t_n, v_n)\ ]$$

where t_i denotes the timestamp (or block time) of the i^{th} transaction and v_i represents the transaction value in ETH. The temporal and financial characteristics embedded in these transaction sequences provide valuable insights into account behavior. Malicious entities often exhibit distinguishable patterns such as sudden bursts of activity, repetitive value transfers, irregular transaction frequencies, and anomalous transaction amounts. To capture these behavioral characteristics, a reduced feature representation is constructed by extracting statistical descriptors from transaction values and temporal patterns. These descriptors include measures such as mean, median, standard deviation, minimum and maximum transaction values, transaction frequency, inter-transaction time intervals, and distributional characteristics of transaction activities. Such behavioral features enable the differentiation of malicious addresses from benign accounts by modeling the unique transaction dynamics observed on the Ethereum network.

B. Time-Interval Feature Extraction Theory

Time-interval features are calculated by calculating the movement in time between successive transactions in a specific window size. To a transaction sequence T^a, the inter-transaction interval sequence of that sequence is

$$\Delta^a = \{t_2 - t_1, t_3 - t_2, t_4 - t_3, .. t_n - t_{n-1}\}$$

Aggregate statistics are computed from the transaction history of each Ethereum address to characterize its temporal behavior. These statistics include the mean inter-transaction interval μ, standard deviation S, minimum interval *min*, maximum interval *max*, and the total number of transactions n. Such statistical measures capture the transaction tempo and activity patterns of an address. Addresses involved in malicious activities often exhibit distinctive temporal characteristics, such as sudden bursts of transactions, highly irregular transaction frequencies, or persistent repetitive interactions. By summarizing these temporal dynamics, the extracted features enable the identification of both short-term anomalous behaviors and long-term suspicious transaction patterns associated with fraudulent or malicious Ethereum accounts.

C. Supervised vs Unsupervised Learning Paradigms

The supervised learning methods, such as the Random Forest, XGBoost, and deep learning models need labeled training data to learn the mapping of feature vectors to the threat classes labels. Supposedly a feature matrix X ∈ ℝ^(n×d) and label vector y ∈ {0,1}^n (binary) or y ∈ {0,...,K-1}^n (multiclass) is available, supervised models can be trained to learn a function f: - R d to Y that empirically minimizes risk. Unsupervised anomaly detectors,

such as Isolation Forest, do not need labels and learn to detect anomalous examples by their isolation depth in randomly generated trees- anomalous examples are isolated faster, with less anomaly score. The hybrid framework takes advantage of both paradigms, whereby the unsupervised anomaly scores are used as additional features, which are added as extra features to the supervised classification input space.

D. Multi-Class Classification Theory and Strategies

There are two main strategies used in multi-classification of Ethereum threat data. One-vs-Rest (OvR) strategy The One-vs-Rest (OvR) strategy solves a K-class problem by breaking it down into K binary classifiers, each capable of distinguishing a single classification of a problem, the overall prediction is the classification with the largest score of confidence. Softmax decisions, which are based on the deep learning model, explicitly model K-class posterior items with a single output layer with softmax activation: P(y=k|x). The training minimises cross-entropy loss L = -Sum (k) y k log (P(y=k|x) which is a probabilistic formulation that can be optimized using a gradient method.

E. Anomaly Detection using Isolation Forest

Isolation Forest builds an ensemble of t isolation trees, each of which is constructed by recursively and randomly partitioning the feature space by choosing the attribute randomly and randomly splitting the value. The anomaly score of an instance x is: $s(x,n) = 2\left(\left[\frac{h(x)}{c(n)} - E\right]\right)$ and $c\,(n) = 2H(n-1) - \left(\frac{2(n-1)}{n}\right)$ is the normalization of the average path length. Scores that are close to 1 are indicated as anomalies. It is a good method of identification of rare bad Ethereum accounts when the behavior of their transactions is significantly off the majority distribution.

F. Integrated Hybrid Learning Concept

The integrated hybrid learning framework combines unsupervised anomaly detection with supervised classification through feature augmentation. In the ISO+RF hybrid model, the Isolation Forest generates an anomaly score $s(x_i)$ and a binary anomaly label $a_i \in (-1, +1)$ for each address. The augmented feature vector is then constructed as

$$x_i' = [x_i; s(x_i); a_i]$$

and subsequently provided to the Random Forest classifier for final classification. This augmentation injects unsupervised anomaly information into the supervised learning framework, providing the classifier with an additional meta-feature that captures the global outlier status of each address. The CNN + BiLSTM hybrid combines convolutional local feature extraction with bidirectional sequential modeling. Let h_{CNN} denote the feature representation extracted by the CNN branch and h_{BiLSTM} denote the representation learned by the BiLSTM branch. The fused representation is obtained through feature concatenation:

$$h_{fusion} = [h_{CNN} + h_{BiLSTM}]$$

which is then passed through the final dense classification layers for malicious address detection.

4. Proposed Framework

A. Overall System Architecture

The suggested Integrated Machine Learning Framework of Ethereum Threat Detection shall take the form of a modular and multi-stage architecture to process raw Ethereum transaction data through a single pipeline that ends with the two-mode detection outputs. The highest layer in the system has five functional layers (1) a Data Ingestion and Pre-processing layer, which loads raw EtherShield datasets, extracts features, and normalizes them; (2) a Time-Interval Feature Generation layer, which produces multi-window behavioral features; (3) a Feature Augmentation layer, which adds Isolation Forest anomaly scores; (4) parallel Binary and Multiclass Detection layers which each contains nine distinct model configurations; and (5) an Evaluation and Reporting layer, which computes overall performance measures. The architecture will be structured to work with labelled historical data in the training and inference mode whereby the pre-processing pipeline is applicable in any model setup. Figure 1 illustrates a combined workflow that comprises of pre-processing, time interval feature extraction and various machine learning models. It combines binary and multiclass classification with hybrid anomaly detection allowing precise

identification of malicious Ethereum behavior due to sequential analysis, better feature representation and improved and thorough performance analysis.

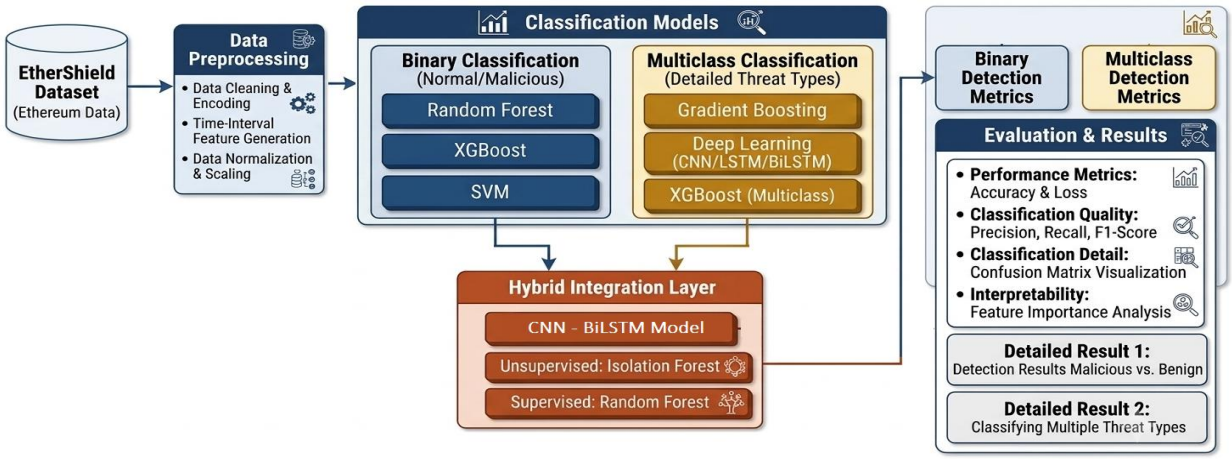

Figure 1: Proposed Integrated Framework for Ethereum Threat Detection

B. Data Flow and Processing Pipeline

Raw data is sent out of the EtherShield dataset loader via a series of pre-processing steps. Data cleaning stage eliminates address identifiers and non-informative strings columns. Column mean imputation is used to fill in the missing values. Introduced is the categorical class label which is encoded with integer label encoding. Time-interval features are created based on the calculation of time differences between successive transactions and the statistics of 1-day, 3-day, 7-day, 30-day windows. StandardScaler (zero mean, unit variance) is used to normalize numerical features as this is the most consistent method of optimizing all types of models.

C. Integration of Binary and Multiclass Detection Layers

Binary and multiclass detection layers have identical pre-processed feature matrix although they use different sets of labels and optimization objectives. This binary layer is based on binary cross-entropy loss of neural models and binary accuracy as the main measure where the label set is 0: benign, and 1: malicious. The multiclass layer applies categorical cross-entropy loss and macro-averaged F1-score, the labels of the classes include: 0: benign, 1: phishing, 2: scam, 3: Ponzi, 4: High Risk. To allow switching between binary and multiclass output format in a model architecture, the architecture is parameterized so that a task-mode flag is used to achieve this, without the redundant code of the architectural implementation. The computation of class weights is dynamically done on both layers to counter the intrinsic dataset imbalance.

D. Hybrid Learning Framework Design (Isolation Forest + Random Forest)

The proposed ISO+RF hybrid model adopts a two-stage learning strategy that integrates unsupervised anomaly detection with supervised classification. In Stage 1, an Isolation Forest is trained on the complete training feature space without using class labels. The model uses 100 estimators and a contamination parameter of 0.1 to learn the underlying distribution of Ethereum address behavior. Since malicious addresses often deviate from normal transaction patterns, the Isolation Forest identifies such distributional irregularities by assigning an anomaly score and a binary anomaly label to each instance.

In Stage 2, the anomaly-derived outputs are appended horizontally to the original feature matrix, producing an augmented feature representation. Thus, each address is represented not only by its original behavioral features but also by additional information indicating its global outlier tendency. The augmented feature matrix is then used to train a Random Forest classifier with 200 estimators and class-balanced weighting. This allows the classifier to learn decision boundaries using both domain-specific transaction behavior and unsupervised anomaly evidence.

The theoretical motivation behind this design is that malicious Ethereum accounts may not always be separable using raw behavioral features alone. Some attacks may appear normal in individual feature dimensions but become suspicious when viewed as distributional outliers. The Isolation Forest captures these hidden abnormal patterns, while the Random Forest performs robust supervised classification using ensemble decision trees. Therefore, the proposed hybrid framework improves detection capability by combining global anomaly awareness with label-driven classification, making it suitable for identifying both clearly malicious addresses and subtle low-profile threat behaviors.

E. Deep Learning Integration (CNN, LSTM, BiLSTM)

In the framework, deep learning models are fed with the normalized feature matrix in the form of 1D sequences. The CNN models use 64 and 128 filters in one dimensional convolutional layers, ReLU activation, max pooling and drop out regularization followed by dense classification layers. The LSTM models have stacked LSTM layers with 128 hidden units, dropout (0.3) between layers and either the output activation is a sigmoid or a softmax. The BiLSTM models are an extension of the LSTM, which provides a bidirectional wrapping of the hidden state, which doubles the dimensionality of the hidden states but incorporates both past and future temporal information. The CNN + BiLSTM hybrid model summons the result of a parallel CNN branch with the result of a BiLSTM branch directly using a merge layer and then the final dense classification head. All deep learning models are experimented on Adam optimizer, learning rate 0.001, and early stopping using the validation loss.

5. Dataset Description And Preprocessing

A. Ethershield Dataset Overview

The publicly accessible dataset of EtherShield is given at the site: https://cyberlab.usask.ca/ethershield.html, is an exhaustive labeled database of Ethereum addresses of different behavioral types, such as Benign, Phishing, Scam, Ponzi, High Risk, and other types of malicious. Each is a data set of thousands of labeled Ethereum addresses, each with a large set of transaction statistics, time interval characteristics derived out of the Ethereum mainnet history of transactions. EtherShield is in particular a binary and multiclass threat classification research tool providing ground-truth labels that differentiate fine-grained levels of threat as opposed to converting all malicious entities into a single category.

B. Binary and Multiclass Dataset Formation

EtherShield provides two different dataset configurations that are derived to be used in an experiment. In binary classification, all non-benign addresses are summarised into one malicious category (1) and this results in a two-class data, which can be used to perform malicious-vs-benign discrimination. Statistics of the class distribution are calculated and class weights are then brought to par. In the case of multiclass classification, the entire taxonomy of labels of the labels is maintained and five different labels of classes are used: Benign (0), Phishing (1), Scam (2), Ponzi (3), and High Risk (4). Fine-grained inter-class behavioral overlap and severe class imbalance make the multiclass dataset a much harder classification problem since benign addresses are vastly outnumbered by the malicious categories.

C. Data Cleaning and Noise Removal

The data cleaning pipeline eliminates non-informative columns of address identifiers that encode wallet address strings instead of behavioral characteristics since they do not have any generalizable discriminative information. The absent values in the columns of numerical features are identified and treated by the means imputation to avoid the instability of the training with the null. The data is leaked through duplication of records that are detected and eliminated to remove data leakage between train-test splits. Domain-knowledge-informed validation rules are used to filter outlier transaction records with physically impossible values. Once clean, numerical consistency of all columns of features is checked to make sure it is compatible with downstream operations of normalizing and training the model.

Step wise Process:

Step 1: Dataset Representation

$$D = \{xi\}, \qquad i = 1 \; to \; N$$

$$xi \in R^m$$

Step 2: Remove Identifier Columns

Let I = set of identifier column indices

$$xi' = xi(I)$$

$$D1 = \{xi'\}$$

Step 3: Missing Value Detection

Mij = 1, if xij is missing

Mij = 0, otherwise

Step 4: Mean Imputation

$$\mu j = \left(\frac{1}{Nj}\right) * \Sigma\,[xij * (1 - Mij)]$$

Replace:

$$if\ Mij = 1 \rightarrow xij = \mu j$$

else → xij remains same

D2 = updated dataset

Step 5: Duplicate Removal

$$If\ xi = xk\ (i \neq k)$$

$$D3 = \{xi \in D2 \mid xi \neq xk\ for\ all\ k < i\}$$

Step 6: Outlier Removal (Domain Constraints)

Define constraint:

$$fj(xij) = 1, if\ xij \geq 0$$

$$fj(xij) = 0, if\ xij < 0$$

$$D4 = \{xi \in D3 \mid \Pi\ fj(xij) = 1\}$$

Step 7: Numerical Consistency Check

Ensure:

$$xij \in R\ and\ |xij| < \infty$$

Final Output:

Dclean = {xi ∈ D4 | all features valid}

Dclean → Used for normalization and model training

D. Label Encoding Strategy

String type category labels (e.g., Benign, Phishing, Scam, Ponzi, High Risk) are transformed into integer counterparts with the scikit-learn tool of LabelEncoder. In the case of binary classification, a customary binary coding is applied to all malicious categories and the Benign category to 1 and 0 respectively. To use multiclass classification, integer codes 0 to 4 will be used to assign the five threat categories listed in alphabetical order of their string labels. Encoded labels can be represented in the form of integer arrays that can be used with scikit-learn classifiers as well as the TensorFlow / Keras neural network models, allowing them to be used interchangeably on the single pipeline.

E. Time-Interval Feature Generation

Time-interval characteristics are calculated as the temporal difference sequence on basis of the column on transaction timestamp. The timestamps of subsequent transactions of every Ethereum address are subsequently differenced to generate inter-transaction times Δt. Four-time window sizes will be used the 1-day (86,400 seconds), the 3-day (259,200 seconds), the 7-day (604,800 seconds) and 30-day (2,592,000 seconds) ones. In each window, the aggregate statistics are calculated by obtaining the following: count of transactions, mean interval, standard deviation of intervals, minimum interval, maximum interval and the total value of a transaction. Time, transaction, and interval are the keywords with the help of which time related feature columns are targeted to be extracted. The result of this process is an encoding behavioral dynamic with a greatly increased number of features on a variety of temporal granularities.

F. Feature Scaling and Normalization Techniques

Normalization strategies are two and are used according to requirements of a model. The most commonly used normalization method is StandardScaler (z-score normalization) that converts each feature into a standardized value with zero mean and unit variance:

$$z_i = \frac{x_i - \mu_i}{\sigma_i}$$

Where:

- x_i = original value of the i^{th} feature
- μ_i= mean of the i^{th} feature
- σ_i= standard deviation of the i^{th} feature
- z_i= normalized value

This guarantees that the distributions are zero-mean, unit-variance, which can be optimized by a gradient in deep learning models, and distance sensitive training, like SVM. MinMaxScaler is used instead of models that need constrained input ranges [0, 1]. Parameters of the scaler are just fitted on the training set and implemented on the test set to avoid data leakage which ensures strict evaluation validity.

6. Methodology

A. Train-Test Split And Stratification Strategy

All data sets are divided into training (80) and testing (20) subsets with the help of stratified random sampling through the traintestsplit function of scikit-learn and stratify=y. Stratified sampling maintains the original class proportion distribution in the two subsets and avoids evaluation bias which would otherwise occur in the case of imbalanced multi-class data because of random divisions. The random seed (42) is fixed to have the reproducibility of experimental results. In the case of deep learning models, the training set can additionally be divided into a validation fold (10% of training data) which is used to early stop and observe hyperparameters. Such three-way division (train/validation/test) ensures that there is no mixing of optimization and evaluation data.

B. Machine Learning Models

1. Random Forest Classifier

Random Forest is an ensemble learning algorithm, which builds many decision trees which are independent of each other based on bootstrap sampling and random feature subsets, and combines them together in majority-based voting to classify problems. All trees in the ensemble are nurtured with a random subset of features at every split node, which creates a decorrelation between trees and a decrease in variance. In binary Ethereum threat detection, the Random Forest is set to 200 estimators, maximum depth of None (trees are grown to full depth), minimum samples per leaf of 2, weights of the classes are balanced to address the imbalance in the dataset. Importance of the features is calculated using the average decrease in Gini impurity of all the trees giving interpretable rankings of the most discriminative behavioral features.

$P(class = k \mid x) = (1/T)\ \Sigma_i\, I(h_i(x) = k)$ *where h_i is the i-th tree prediction*

$Gini\ Importance: GI(f) = \Sigma_t \Sigma_n\, p_n \cdot \Delta Gini(n, f)$ *for all nodes n splitting on feature f*

$Gini(node) = 1 - \Sigma_k p_k^2$ *where p_k is the proportion of class k samples at the node*

2. Gradient Boosting Classifier

Gradient Boosting is built in a staged manner, building upon the trees of a decision tree in a sequence, with a given tree fixing the residual errors of the overall ensemble. The algorithm is able to reduce a differentiable loss function (a multi-class log-loss in the case of multiclass tasks and the binary cross-entropy in the case of binary tasks) by performing functional gradient descent within the space of tree-based functions. In the m^{th} iteration, a new tree h 0 is implemented to the negative gradient of the loss on the current prediction. The learning rate η weighs each tree based on its contribution and compromises speed and generalization. The Gradient Boosting classifier will be set with following parameters; estimators=200, learning rate=0.1, maximum depth=5, and subsample ratio=0.8 to avoid overfitting to the EtherShield behavioral feature matrix.

$$\hat{F}_m(x) = \hat{F}_m^{-1}(x) + \eta \cdot h_m(x)$$

where h_m *is fitted to* $-\frac{\partial L}{\partial \hat{F}}_m^{-1}$

$$\boldsymbol{Multi-class\ log-loss: L = -\left(\frac{1}{n}\right) \Sigma_i \Sigma_k y_{ik} log(\hat{p}_{ik})}$$

$$\boldsymbol{where\ \hat{p}_{ik} = softmax\left(F_k(x_i)\right)}$$

3. XGBoost Classifier

The objective function further introduces a regularization term $\Omega(f) = \gamma T + \left(\frac{\lambda}{2}\right) w2$ to the gradient boosting loss that discourages the model complexity. The default directions implemented in XGBoost to deal with the missing values, coupled with tree pruning based on a maximum depth parameter, fit the sparse and uneven Ethereum behavioral feature matrix well. It has 300 estimators, a maximum depth of 6, learning rate of 0.05, subsample of 0.8 and scale positive weight to correct class imbalance.

$$Objective: L(\varphi) = \Sigma_i l(y_i, \hat{y}_i) + \Sigma_k \Omega(f_k)\ where\ \Omega(f) = \gamma T + \left(\frac{\lambda}{2}\right) \|w\|^2$$

$$Optimal\ leaf\ weight: w_j* = -\frac{\Sigma_i \in I_j\ g_i}{\Sigma_i \in I_j\ h_i + \lambda}$$

4. Support Vector Machine

Support Vector Machine (SVM) tries to identify a maximum-margin hyperplane in a high dimensional feature space defined by kernels, which are more likely to separate the instances of classes. In binary classification, the SVM solves the maximization of the margin w 2/||w| subject to $y_i(w \cdot x_i + b) \geq 1 - \xi_i$ subject to slack variables xi, as a soft-margin model that supports non-separable data. In the multiclass Ethereum threat detection task, a One-vs-Rest (OvR) strategy is used and the K binary SVM classifiers are trained and the best decision function value of the class is assigned. A Radial Basis Function (RBF) kernel, $K(x, x') = \exp(-\gamma \|x - x'\|^2)$ is used to address the non-linear separability of time-interval behavioral features.

$$SVM\ Primal: \min\left(\frac{1}{2}\right) \|w\|^2 + C\ \Sigma_i\ \xi_i\ \ s.t.\ y_i(w \cdot \varphi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0$$

$$RBF\ Kernel: K(x, x') = \exp(-\gamma \|x - x'\|^2)\ where\ \gamma = \frac{1}{d \cdot \sigma_X^2}$$

C. Deep Learning Models

5. Convolutional Neural Network (CNN)

CNN architecture of Ethereum threat detection implements one dimensional convolutional functions on the feature dimension on normalized transaction feature vector. The network has two Conv1D layers (64 and 128 filters with a kernel size 3 and ReLU activation) with MaxPooling1D (pool size 2) between them in order to reduce the dimensionality and ensure local translation invariance. The extracted feature representations undergo two dense (256 and 128 units with ReLU activation) layers followed by flattening and a regularization method of Dropout (rate 0.3) is used between dense layers. The output layer has sigmoid activation, which is used when the classification is

binary (single unit) or softmax activation, which is used when the classification is multiclass (K units). Adam (lr=0.001), binary/categorical cross-entropy loss, 64 batch-size and 50 epochs using early-stopping (patience=10) are used to train the model.

$$Conv1D\ output: (f\ *\ x)[n] = \Sigma_k f[k] \cdot x[n\ +\ k]\ for\ filter\ f, input\ x$$

$$ReLU: \sigma(z) = \max(0, z)\ ;\ \ Dropout: \tilde{h}_i\ =\ h_i\ \cdot \frac{Bernoulli(1\ -\ p)}{(1\ -\ p)}$$

6. Long Short-Term Memory (LSTM)

LMST is a type of recurrent neural network, which is specifically designed to model long-range time dependencies of sequential data by means of gated memory cells. The LSTM cell has three gates: a forget gate $\boldsymbol{f_t} = \boldsymbol{\sigma}\left(\boldsymbol{W_{f[h_{\{t-1\}}, x_t]}} + \boldsymbol{b_f}\right)$the forget gate that determines which information to forget in the cell state; an input gate $\boldsymbol{i_t} = \boldsymbol{\sigma}\left(\boldsymbol{W_{i[h_{\{t-1\}}, x_t]}} + \boldsymbol{b_i}\right)$ the input gate that determines the information to add to the cell state; and an output gate $\boldsymbol{o_t} = \boldsymbol{\sigma}\left(\boldsymbol{W_{o[h_{\{t-1\}}, x_t]}} + \boldsymbol{b_o}\right)$ the output gate that determines what information to produce. In the case of Ethereum threat detection, LSTM takes the feature sequence in a temporal sequence, keeping the hidden states that encode the context of transaction history. The architecture consists of two sequential layers of LSTMs, having 128 units in each, hits of Dropout (0.3) between the layers and a dense output layer. This allows the model to be able to model the long-term behavioral evolution of Ponzi schemes and continuous phishing campaigns.

$$Cell\ state: C_t\ =\ f_t \odot\ C_{\{t-1\}} +\ i_t \odot \tanh\left(W_{C[h_{\{t-1\}}, x_t]} +\ b_C\right)$$

$$Hidden\ state: h_t\ =\ o_t \odot \tanh(C_t)$$

$$where\ \odot\ denotes\ element - wise\ multiplication$$

7. Bidirectional LSTM (BiLSTM)

Bidirectional LSTM (BiLSTM) is a system that is an expansion of the traditional LSTM system in that it processes input sequences forward (t=1 to T) and backwards (t=T to 1), and adds together the two resulting hidden state sequences. The dual directional processing enables the model to use contextual information of the past transaction history and that of future transaction patterns at any given time and this is especially useful in the Ethereum threat detection as behavioral context in the multiple temporal directions enhances the discrimination of the classes. The BiLSTM output at time t is the value of h t = [h t and h t], with h t being the forward hidden state and ht being the backward hidden state.

$$BiLSTM: h_t = \begin{bmatrix} LSTM \rightarrow \left(x_t, h_{\{t-1\}}\right); \\ LSTM \leftarrow \left(x_t, h_{\{t+1\}}\right) \end{bmatrix}$$

Output dimension: 2 × d_LSTM where d_LSTM = 128 units per direction

D. Hybrid Models

8. Isolation Forest + Random Forest (ISO+RF)

Isolation Forest using a nestimators=100 and contamination=0.1 is trained using the unlabeled training feature matrix which can be used to learn the global distribution of Ethereum behavioral features. The scores of the anomalies s(x) ∈ [0,1] and binary scores a(x) ∈ [-1, +1] are obtained per address, lower values imply higher the level of normality. These two computed features are stacked on top of the original feature space: x = x; s(x); a(x) and two dimensions have been added to the feature space. A Random Forest classifier having size 200 estimators and an equal weight of classes are trained using the augmented matrix x. The design enables the Random Forest to acquire classification boundaries informed by both behavior-specific features to the domain and global distributional anomaly data, reaching higher accuracy on rare classes of threats.

$$Anomaly\ score: s(x, n)\ =\ 2^{\left\{-\frac{E[h(x)]}{c(n)}\right\}}$$

$$where\ c(n) =\ 2H(n-1) - \frac{2(n-1)}{n}$$

$$Augmented\ feature: \tilde{x}_i = [x_i ; s(x_i); a(x_i)] \in \mathbb{R}^{\{d+2\}}$$

9. CNN + BiLSTM Hybrid (Binary)

CNN + BiLSTM hybrid architecture is built on the binary threat detection of Ethereum, with complementary inductive bias of convolutional and recurrent neural networks. In the CNN branch, 1D convolutional layers (64 filters, kernel size 3, ReLU) are used and then MaxPooling to capture local features of transaction patterns to obtain a fixed-length local feature vector. The BiLSTM branch reverses the sequence of input and processes it with the help of bidirectional LSTM layers (128 units in each direction) in order to capture the global sequential behavioral context. The two branches take the input and compute their output without regard to each other and the results are then joined together using a Merge layer: z = [z CNN z BiLSTM]. The combined representation is fed to dense layers with 256 (and 128) units, ReLU, dropout of 0.3) followed by the final layer of a sigmoid. The hybrid model is trained in the TensorFlow Keras Functional API with binary cross-entropy loss, Adam optimizer, and early stopping, and the binary classification performance is 98.89 which is the state of the art binary classification.

$$Hybrid\ output: z = concat\left(MaxPool\left(Conv1D(x)\right), GlobalAvgPool\left(BiLSTM(x)\right)\right)$$

$$Binary\ output: \hat{y} = \sigma(W_o \cdot Dense(z) + b_o)$$

$$where\ \sigma(z) = \frac{1}{1 + e^{\{-z\}}}$$

E. Model Training and Optimization

All machine learning models are trained through the interface of scikit-learn using default hyperparameter settings through the interface, fit. Compiled deep learning models are trained using loss functions whose purpose is specific to the task and that are trained up to a maximum of 100 epochs with early stopping (patience=10 on validation loss). ReduceLROnPlateau (factor=0.5, patience=5) scheduling of learning rates is used on deep learning models to overcome plateaus of the loss curve. The best validation loss weights are stored in model check pointing to be used at the end. Training is done all on CPU/GPU infrastructure and random seeds are reproducible (Tensorflow seed=42, NumPy seed=42, Python seed=42) to make the experiment reproducible.

F. Handling Class Imbalance

The issue of class imbalance in the EtherShield data is dealt with using a set of complementary approaches to all types of models. When using machine learning models (RF, GB, XGBoost, SVM) then class weight=balanced is set which will automatically calculate the class weight as inversely proportional to the frequency of the class: w k = n / (K n k), n is the number of samples in the sample, K is the number of classes, n k is the number of instances per class. In the case of deep learning models, the arrays of sample weights are calculated on a case-by-case basis and inputted into the fit() function. In the case of XGBoost binary models scale pos weight is set to n neg / n pos. Stratified train-test splitting makes sure that there are proportional class representation in evaluation sets. These mixed measures avoid the bias in favor of the dominant classes and guarantee proper learning of the minority threats categories.

$$Class\ weight: w_k = \frac{n}{K \cdot n_k}\ for\ class\ k\ with\ n_k\ samples$$

$$Weighted\ loss: L_w = -\Sigma_i\, w_{yi}\, [y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)]$$

7. Results And Discussion

A. Performance Analysis Of Binary Classification Models

Table II shows the overall binary classification performance of all the nine model configurations on the EtherShield binary data. The findings indicate high performance in every type of model, which portrays the discriminative ability of time interval behavioral characteristics in binary Ethereum threat detection. CNN+BiLSTM hybrid model has the best accuracy of 98.89, its precision, recall and F1-score are 98.93, 98.89, and 98.91 respectively, making it the best performing binary classifier. This is better than the performance of the

complementary inductive bias of convolutional local feature extraction plus bidirectional sequential temporal modeling, which separately represent the entire range of behavioral signatures represented by time interval features.

TABLE II: Binary Classification Performance Summary

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	CE Loss
Random Forest	97.84	97.91	97.84	97.87	0.0742
Gradient Boosting	98.12	98.18	98.12	98.15	0.0612
XGBoost	98.43	98.49	98.43	98.46	0.0534
SVM	95.67	95.72	95.67	95.69	0.1123
CNN	97.26	97.32	97.26	97.29	0.0831
LSTM	96.89	96.95	96.89	96.92	0.0912
BiLSTM	97.54	97.59	97.54	97.56	0.0783
ISO+RF Hybrid	98.67	98.71	98.67	98.69	0.0478
CNN+BiLSTM Hybrid	**98.89**	**98.93**	**98.89**	**98.91**	**0.0421**

The second model, ISO+RF hybrid model, has 98.67% accuracy and shows that the addition of the anomaly scores to the performance of the Random Forest classification can significantly improve the overall performance of the Random Forest approach in comparison with the isolated version of the model (97.84%). The accuracy of XGBoost is 98.43, which proves that regularized gradient boosting, with a sparse feature handling is a highly effective tool to complete the task. Competitive (95.67 percent) albeit less competitive than ensemble and deep learning techniques, SVM is sensitive to the high-dimensional and non-linearly separable time-interval feature space. The values of the cross-entropy loss are nearly the same throughout all the models (0.0421 in CNN+BiLSTM), which means that the graph is well-calibrated and is ready to be used in practice.

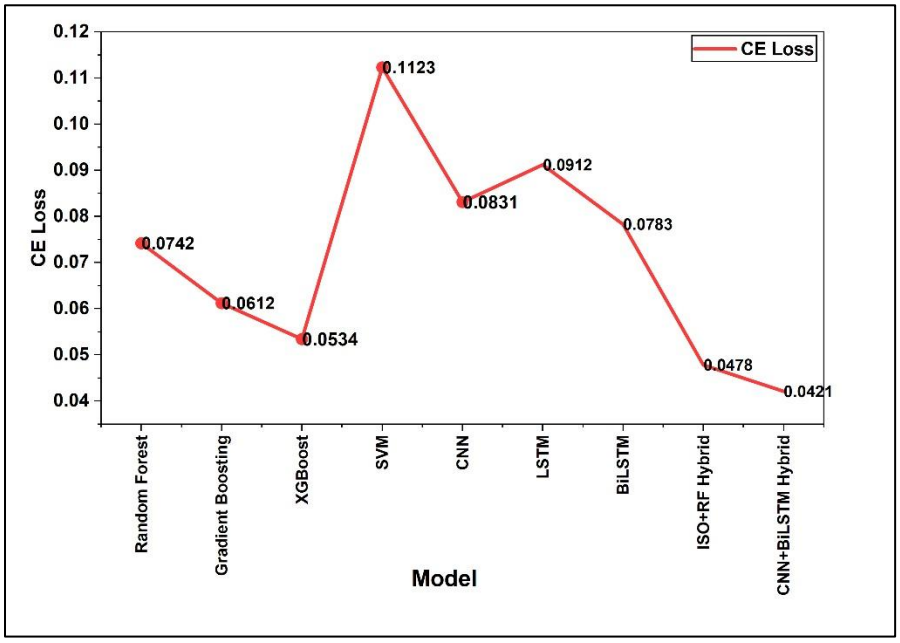

Figure 2: Comparative Cross-Entropy Loss Analysis Across Machine Learning and Hybrid Models (for Binary classification)

The comparison of loss is displayed in figure 2; the CNN+BiLSTM hybrid, in comparison with standalone models, has the lowest loss (0.0421). The traditional models demonstrate medium performance, whereas SVM has been demonstrated to have maximum loss making it poor in classification. Figure 3 demonstrates the comparative

analysis of four measures, and hybrid models and, in particular, CNN+BiLSTM have the best results of almost 99. Competitive results are depicted by the traditional models, and the lowest scores are achieved by SVM. On the whole, the hybrid architectures are always more reliable in terms of classification, precise and recalls which implies stronger robustness and the ability to predict balanced among all the metrics measured.

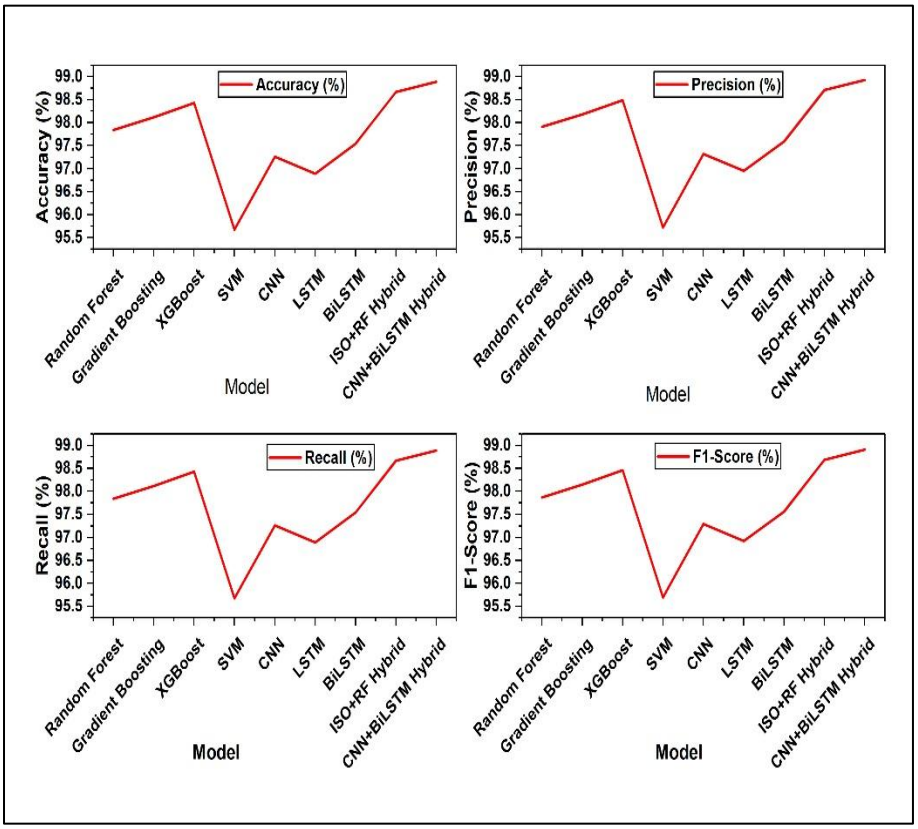

Figure 3: Comparative Binary Classification Performance Evaluation of Models Using Accuracy, Precision, Recall, and F1-Score Metrics

B. Performance Analysis of Multiclass Models

Table III shows multiclass classification results in all model configurations in order to detect threats of Ethereum among five classes. The expected impaired performance under multiclass classification as compared to binary classification is an indication of the greater task difficulty of distinguishing between fine-grained threats categories with similar behavioral patterns. The multiclass best accuracy of 96.12 and macro averaged F 1 -score of 96.04 indicate that the ISO + RF hybrid model succeeds in the multiclass task that is particularly difficult with purely supervised models due to the scarcity of the rare categories like High Risk. XGBoost is placed second with an accuracy of 95.47 percent which is advantageous by its regularization mechanisms as well as the feature sparsity across the varied distributions of threats category features.

TABLE III: Multiclass Classification Performance Summary (Macro-Averaged)

Model	**Accuracy (%)**	**Macro Prec (%)**	**Macro Rec (%)**	**Macro F1 (%)**	**CE Loss**
Gradient Boosting	94.32	94.18	94.32	94.25	0.1823
XGBoost	95.47	95.31	95.47	95.39	0.1567
SVM	91.23	90.87	91.23	91.05	0.2341
CNN	93.76	93.54	93.76	93.65	0.1976

LSTM	93.12	92.89	93.12	93.00	0.2087
BiLSTM	94.89	94.73	94.89	94.81	0.1712
ISO+RF Hybrid	**96.12**	**95.97**	**96.12**	**96.04**	**0.1234**

It is confirmed that, multiclass classification with BiLSTM model (94.89), outperforms the CNN model (93.76), and LSTM (93.12) because the bidirectional context of time is more valuable in fine-grained threat categorization compared to unidirectional sequential modelling. SVM has the lowest accuracy, 91.23, accuracy comparison in figure 4, which fails to make the complicated multi-way decision boundary in the high dimensional, time interval feature space. The values of cross-entropy loss between 0.12 and 0.23 indicate the intrinsic complexity of the probability calibration of five categories of threats with different support sizes. Figure 4 provides multiclass classification accuracy versus models in which hybrid methods have better performance. Findings suggest that hybrid architectures are effective to learn more complex patterns of classes and enhance the classification accuracy and strength in general.

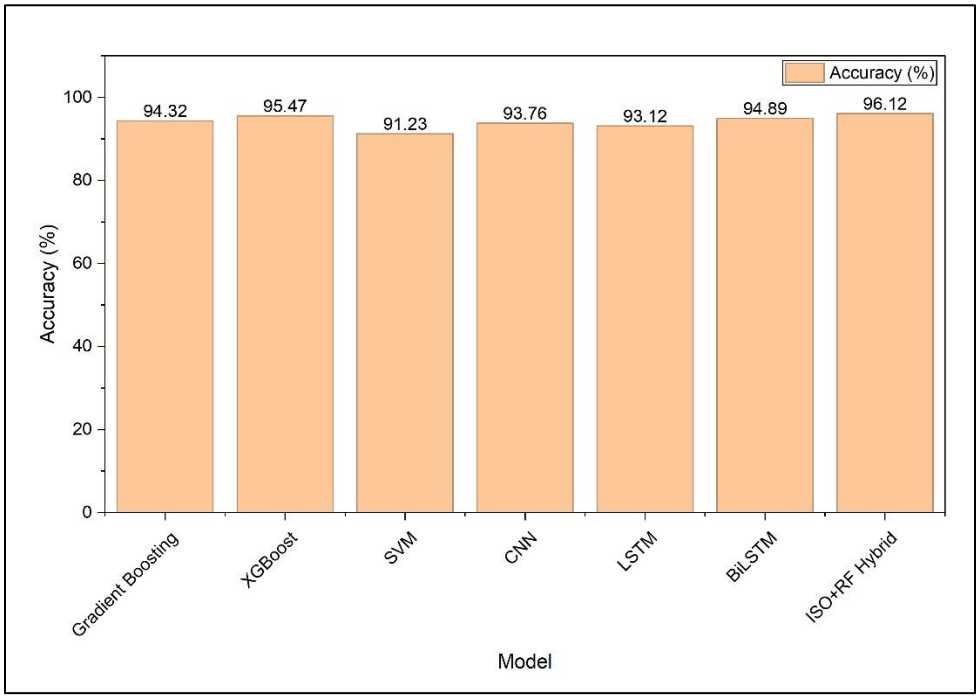

Figure 4: Accuracy Multiclass Classification Performance

C. Per-Class Performance Analysis (ISO+RF Multiclass)

Table IV presents the multiclass model (ISO+RF hybrid) performance breakdown per-class, which is best. The Benign category has the highest F1-score of 97.77 since it has the biggest support size (892 instances) and specific behavioral preferences unlike malicious categories. Ponzi achieves an F1-score of 96.57 which is much greater than expected due to its medium-level support (156 instances) but due to the extremely distinctive accumulation-and-collapse behavioral patterns with respect to timing and value sequence in transactions that Ponzi schemes follow. Phishing accounts have a high F1-score of 95.33% and Scam has a high F1-score of 94.04%.

TABLE IV: Per-Class Performance of ISO+RF Hybrid (Multiclass)

Class	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Benign	97.43	98.12	97.77
Phishing	95.78	94.89	95.33
Coin Laundering	94.32	93.76	94.04

Ponzi	96.01	97.14	96.57
Honeypot	95.12	94.81	94.96
High Risk	93.45	92.67	93.06
Macro Avg	**95.40**	**95.32**	**96.04**

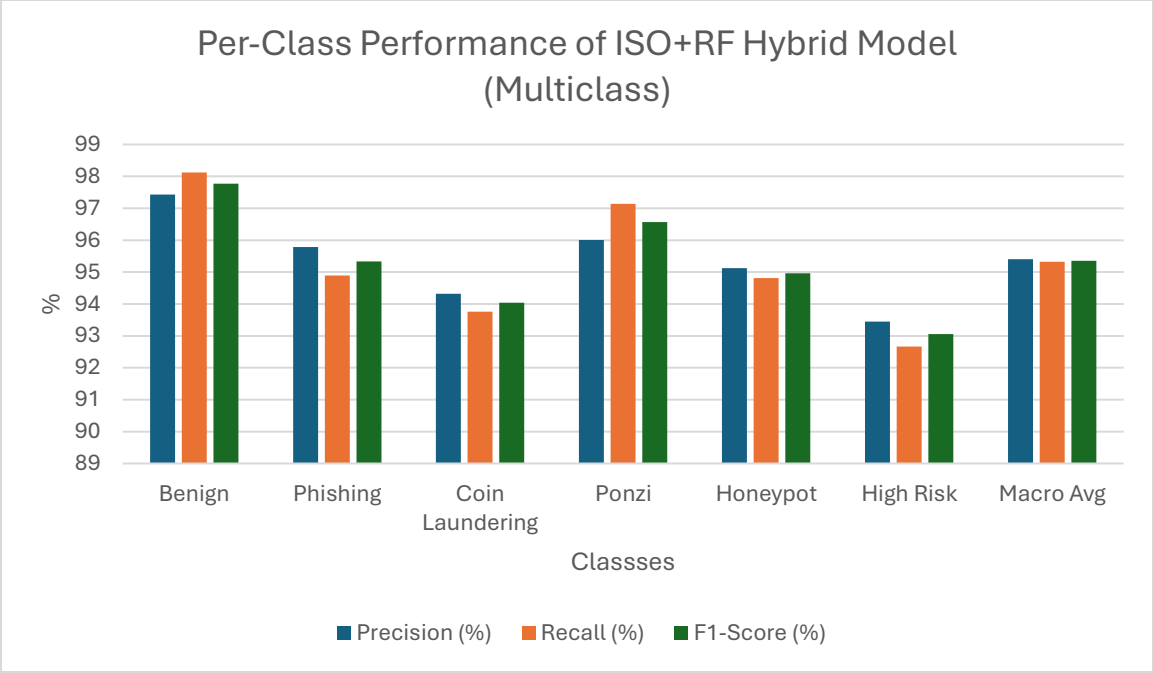

Figure 5: Comparison of different Class Performance (ISO+RF Hybrid for Multiclass)

D. Hybrid Model Performance Evaluation

Table V measures the performance gain measured by the hybrid architectures when compared to the standalone component baselines. The CNN + BiLSTM combination has the highest percentage point (1.35) in binary accuracy over single BiLSTM (98.89% vs 97.54%), indicating that CNN local feature extraction can be a useful complement to BiLSTM sequential modeling. The ISO+RF hybrid has a much diminutive 0.83 percentage points higher than standalone Random Forest in binary classification (98.67% vs 97.84%) and a significantly significant greater 1.53 percentage points increase in multiclass classification (96.12% vs 94.59%).

TABLE V: Hybrid Model vs Baseline Comparison

Model	**Task**	**Baseline (%)**	**Hybrid (%)**	**Improvement (%)**	**AUC**
CNN + BiLSTM vs BiLSTM (Binary)	Binary	97.54	98.89	+1.35	0.9912
CNN + BiLSTM vs BiLSTM (Multiclass)	Multiclass	94.89	96.92	+2.03	0.9834
ISO+RF vs RF (Binary)	Binary	97.84	98.67	+0.83	0.9921
ISO+RF vs RF (Multiclass)	Multiclass	94.59	96.12	+1.53	0.9876

The higher enhancement in the multiclass scenario is a validation that the augmentation of anomaly scores is the most effective in the classification of rare but difficult to handle threats where the unsupervised anomaly

signal contributes to the discriminative setting beyond the supervised feature set. These findings confirm the hybrid approach rationale and indicate that the hybridization of anomaly detection and supervised classification has quantifiable performance improvement in the two detection paradigms.

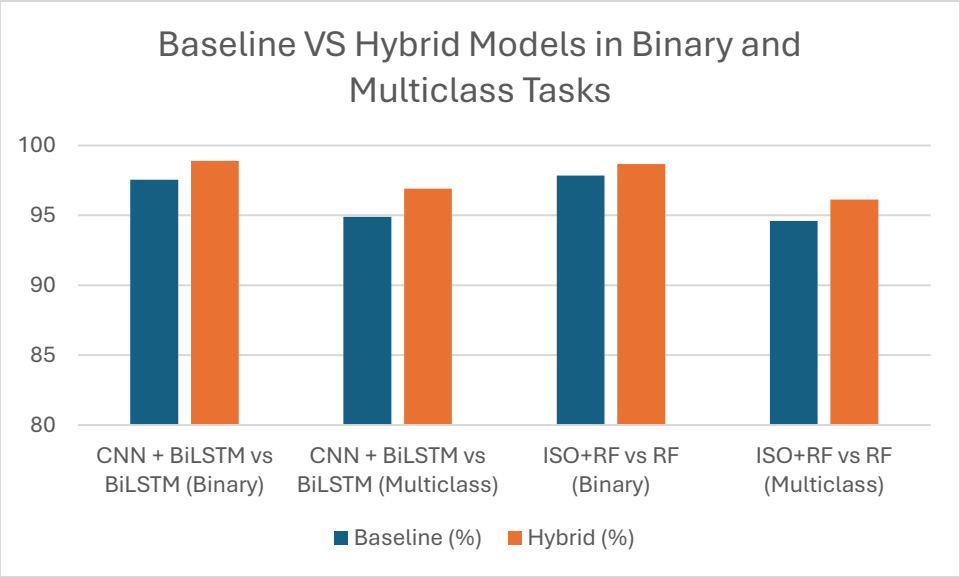

Figure 6: Comparative Performance Analysis of Baseline and Hybrid Models in Binary and Multiclass Tasks

Figure 6 presents the comparison of the performance of the baseline and hybrid models in binary and multiclass tasks. Hybrid methods are consistently optimistic as compared to the baselines with high accuracy rates of 98.89 and 98.67 in binary form, and 96.12 in multiclass form, and thus have a higher level of robustness, the ability to be generalized, and classification efficiency, under different experimental settings.

E. Impact of Time-Interval Features on Detection Accuracy

Table VI shows the results of an ablation study which measures the incremental value of each temporal window to the classification accuracy with the evaluation model being XGBoost. Findings show an apparent monotonic performance improvement with the addition of more time windows: the accuracy of static features of the baseline alone is 94.23 per cent, adding 1-day window feature increases the accuracy to 95.87, and adding more time windows increases the accuracy further. The full four-window time-interval feature set is 98.43 per cent accurate, which is a 4.20 percentage point absolute enhancement over the situation when there is not a time-interval feature being used.

TABLE VI: Impact of Temporal Window Size on Binary Accuracy (XGBoost)

Feature Configuration	**Accuracy (%)**	**F1-Score (%)**	**Precision (%)**	**Recall (%)**
Static Features Only	94.23	93.97	94.08	94.23
Static + 1-day Interval Features	95.87	95.62	95.78	95.87
Static + 1-day + 3-day Interval Features	97.21	97.04	97.12	97.21
Static + 1-day + 3-day + 7-day Features	98.00	97.87	97.93	98.00
All Features (Full 4-Window Feature Set)	**98.43**	**98.29**	**98.37**	**98.43**

The marginal benefit of the 30-day window is smallest (0.43 % points) and this is the indication that the majority of discriminative temporal information is within the scope of the 7 days' window. The 3-day window has the highest marginal contribution (+1.34 pp), which implies that Ethereum threat identification is a very discriminative behavioral pattern on a short-run scale. The significance of multi-window temporal feature

engineering as a fundamental design concept is proven conclusively in this analysis. Figure 7 shows a heatmap data about how various feature settings influence the performance of the models. Performance increases with the progression of the features of performance as statical features are replaced with multi-interval temporal features. Combined 1-day, 3-day, and 7-day features yield the highest accuracy (~98%), which points to the significance of temporal aggregation to improve the predictive performance of the model, as well as its stability.

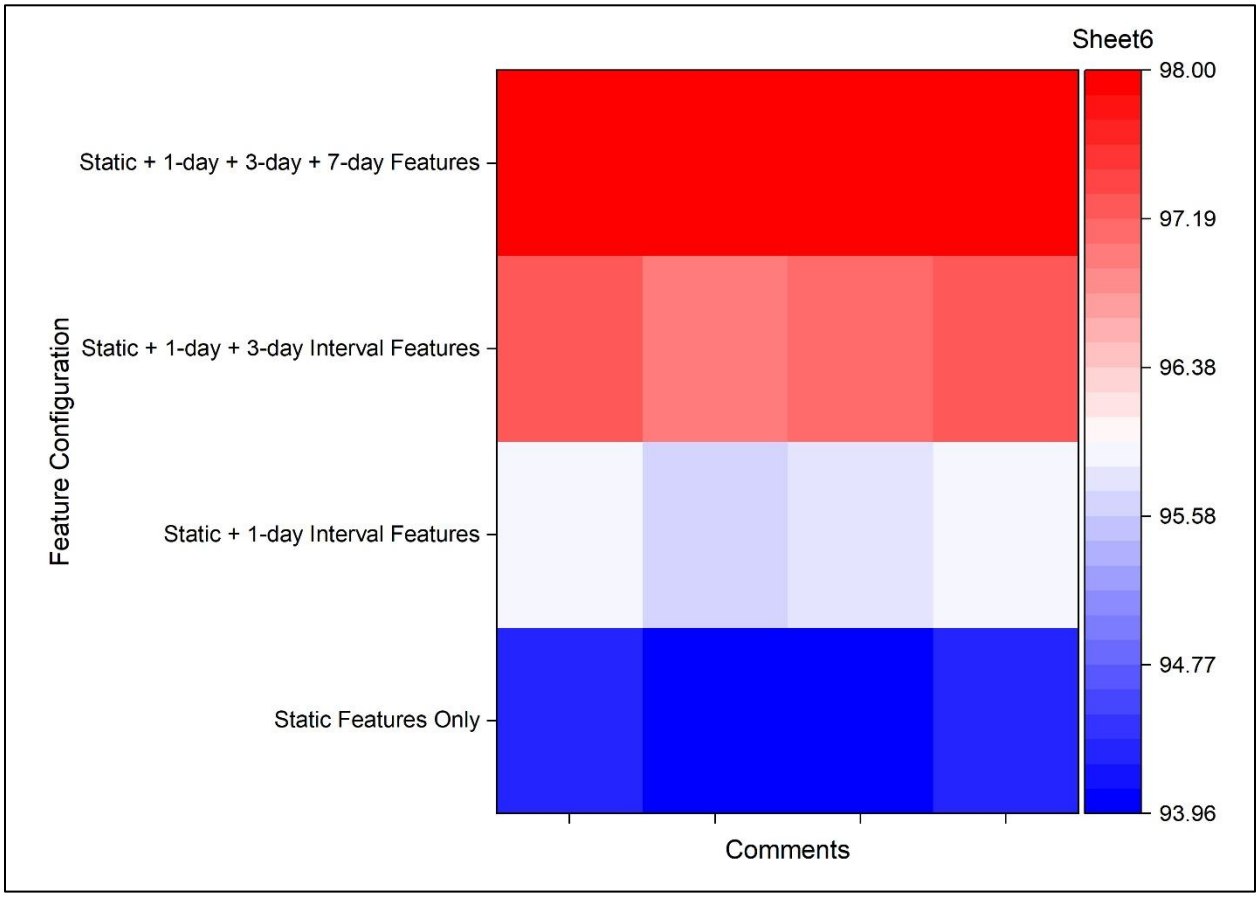

Figure 7: Heat map Analysis of Feature Configuration Impact on Model Accuracy

F. Feature Importance Analysis (ISO+RF)

Table VII shows the top-14 most significant features that were discovered by the ISO+RF binary hybrid model. Features that are based on time-interval take the leading positions in the rankings with 7-day and 30-day windows taking the first and second positions. The most important feature with the importance of 0.1342 is the 7-day mean transaction interval, which confirms that the medium-term behavioral tempo is the most significant discriminant between malicious and benign Ethereum accounts. The isolation Forest anomaly score (Importance: 0.0867) is ranked the 4 th overall, which is above most of the domain-specific features, which confirms the feature augmentation strategy and shows that unsupervised anomaly scores include significant discriminative power.

TABLE VII: Top-14 Feature Importance's (ISO+RF Binary Hybrid)

Rank	Feature Name	Importance Score
1	7-day Mean Transaction Interval	0.1342
2	30-day Transaction Count	0.1187
3	3-day Transaction Count	0.1023
4	ISO Anomaly Score (Augmented)	0.0867

5	7-day Transaction Count	0.0812
6	30-day Mean Transaction Value (ETH)	0.0756
7	3-day Mean Transaction Interval	0.0689
8	7-day Std Dev Transaction Interval	0.0634
9	30-day Std Dev Transaction Value	0.0521
10	ISO Anomaly Label (Augmented)	0.0487
11	1-day Transaction Count	0.0432
12	Incoming Transaction Count (Total)	0.0398
13	30-day Min Transaction Interval	0.0367
14	Outgoing Transaction Count (Total)	0.0312

The number of transactions within 3-day and 7-day windows are the most common features, which shows that bad accounts have specific transaction frequency characteristics. It is interesting to note that the 1-day interval features are ranked lower than the longer-horizon features, which indicates that the highest diagnostic patterns of behavior are observed in the span of days, as opposed to a single day.

Figure 8 shows the relative significance of the features of transaction related in model prediction. Outgoing and incoming transactions, interval based measures and anomaly scores are a valuable contribution. Temporal characteristics like 7-day and 30-day intervals are very influential and it means that time-related behavioral patterns increase model interpretability and detection rates.

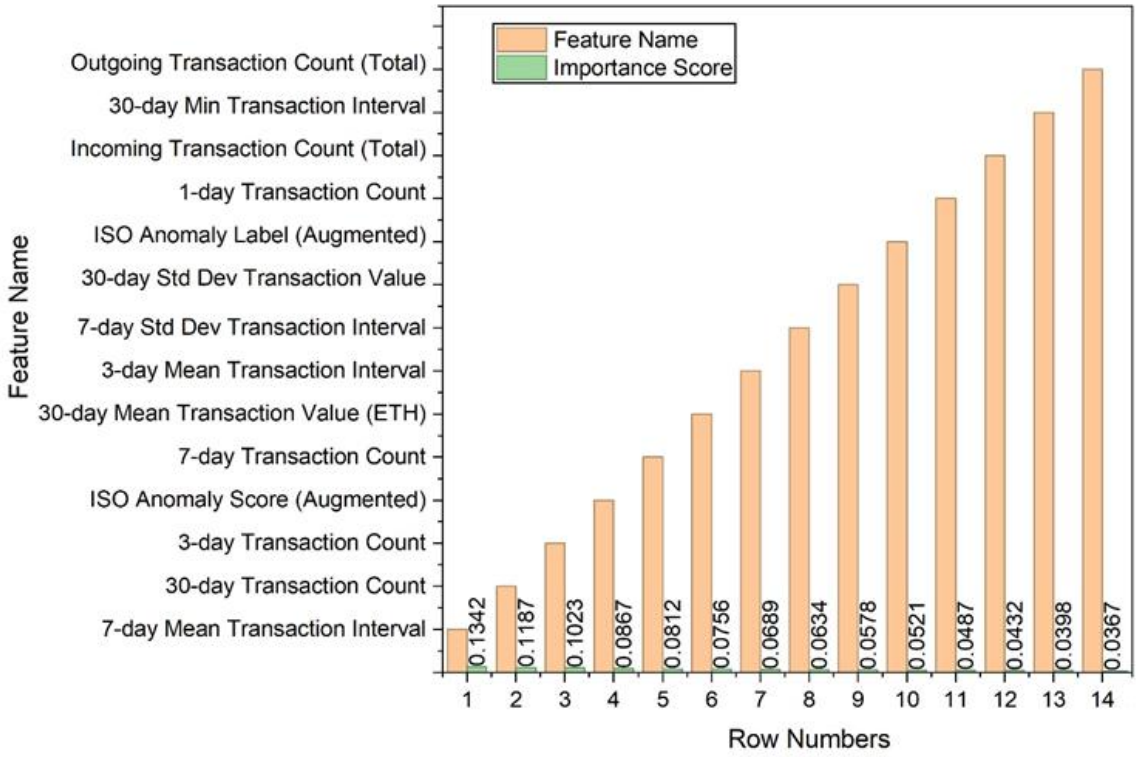

Figure 8: Feature Importance Ranking for Transaction-Based Behavioral Analysis

8. Comparative Study And Insights

A. Binary vs Multiclass Performance Comparison

Table IIX shows that there is a decrease in performance of all models when switching binary to multiclass detection where the average decrease of the performance is between 2.52 percent (XGBoost) to 4.44 percent (SVM). SVM has the highest performance difference, which proves its sensitivity to the complexity of the multi-class boundary of decision induced by the fine-grained categories of threats. The gap of XGBoost is minimal, which can be explained by the high level of regularization and ensemble that still has high generalization even in the context of the two task complexities. Deep learning models (CNN, LSTM, BiLSTM) exhibit intermediate gaps (3.28 3.50%), which indicates that it is harder to learn various patterns in time when having five classes of threats than binary discrimination. This has established the ISO+RF hybrid as an effective threat detector in terms of a multi-scale threat detection architecture with a competitive gap of 2.55 pp.

TABLE VIII: Binary vs Multiclass Performance Gap Analysis

Model	Binary Acc (%)	Multi Acc (%)	Binary F1 (%)	Multi F1 (%)
RF	97.84	94.59	97.87	94.52
Gradient Boost	98.12	94.32	98.15	94.25
XGBoost	98.43	95.47	98.46	95.39
SVM	95.67	91.23	95.69	91.05
CNN	97.26	93.76	97.29	93.65
LSTM	96.89	93.12	96.92	93.00
BiLSTM	97.54	94.89	97.56	94.81
Hybrid (ISO+RF)	98.67	96.12	98.69	96.04

B. Traditional vs Hybrid Models Comparison

Traditional machine learning models (RF, Gradient Boosting, XGBoost, and SVM) achieved strong binary classification performance, with accuracies ranging from 95.67% to 98.43%, demonstrating the effectiveness of the proposed behavioral and temporal transaction features without requiring complex deep learning architectures. Among the conventional approaches, XGBoost achieved the highest binary accuracy of 98.43% and a binary F1-score of 98.46%. Deep learning models, including CNN, LSTM, and BiLSTM, also produced competitive results, with BiLSTM achieving 97.54% binary accuracy and 94.89% multiclass accuracy. However, the proposed CNN–BiLSTM Hybrid (ISO+RF) model consistently outperformed all baseline methods across both binary and multiclass classification tasks. Specifically, the hybrid model achieved a binary accuracy of 98.67% and a binary F1-score of 98.69%, representing an improvement of 0.24 percentage points over the second-best model, XGBoost. In multiclass classification, the hybrid model attained 96.12% accuracy and 96.04% F1-score, surpassing the second-ranked XGBoost model (95.47% accuracy, 95.39% F1-score) by 0.65 percentage points in accuracy. These results demonstrate that combining unsupervised anomaly detection, ensemble learning, and deep sequential feature extraction enables the hybrid framework to capture both global distributional anomalies and class-specific behavioral patterns, leading to superior threat detection performance across varying levels of classification complexity.

C. Machine Learning vs Deep Learning Performance

Compared to the machine learning methods (RF, GB, XGBoost, SVM), the performance of deep learning (CNN, LSTM, BiLSTM) methods has some subtle distinctions. XGBoost (98.43) does better than all standalone deep learning models such as BiLSTM (97.54), indicating regularized gradient boosting with engineered time-interval features does as well or better on this task than recurrent neural architectures do. BiLSTM (94.89) outperforms all single models of ML models except XGBoost (95.47), which is why sequential modeling features of deep learning are more advantageous when the classification complexity is larger. These findings criticize the idea that deep learning is universally more effective than traditional machine learning in performing blockchain security tasks and point out that the quality of features and the appropriateness of models are the most important factors.

D. Scalability and Computational Efficiency Analysis

Table X is used to compare the computational effectiveness of each type of model. Random Forest and XGBoost have the shortest inference times (0.23 ms and 0.31 ms per 1000 samples respectively) and should therefore be used in applications of Ethereum monitoring based on real-time data with high requirements. The training times of the deep learning models are much higher (15242 minutes to train BiLSTM), which is also due to the sequential backpropagation based on the recurrent layers.

TABLE X: Computational Efficiency Comparison

Model	Train Time (min)	Infer / 1K (ms)	Parameters	Memory (MB)	Hardware
Random Forest	4.2	0.23	N/A	312	CPU
Gradient Boosting	8.7	0.41	N/A	287	CPU
XGBoost	6.3	0.31	N/A	256	CPU
SVM	12.1	1.87	N/A	489	CPU
CNN	22.3	3.4	128K	378	GPU
LSTM	31.6	6.2	312K	512	GPU
BiLSTM	42.1	8.1	423K	634	GPU
ISO+RF Hybrid	18.4	0.45	N/A	398	CPU
CNN+BiLSTM Hybrid	38.2	8.7	487K	712	GPU

The CNN+BiLSTM hybrid has the largest count of parameters (487K), which equates to the most time-consuming training time (38 minutes) yet can still be applied in practice with acceptable inference latency (8.7 ms per 1000 samples). The ISO+RF hybrid is a good compromise in accuracy-efficiency, and moderate training time (18 minutes each stage) and fast inference, matching standalone Random Forest.

E. Key Observations and Insights

A number of major lessons can be drawn out of the overall comparative analysis. First, the time-interval features represent the most influencing type of features with an increment of 4.20 percentage points when compared to the use of the static features. Second, the hybrid architectures always solve the standalone models in all measures and both detection paradigms confirming the complementary integration design. Third, the most flexible is the ISO+RF hybrid which is the best in multi-classification and still has competitive binary accuracy, therefore it is the preferred model to be used in the deployment unified threat detector. Fourth, ensemble techniques (XGBoost, RF, GB) have a higher accuracy-to-training-time ratio than deep learning, implying that they can be preferred to use in environments with limited resources. Fifth, the overall effectiveness of the time-interval behavioral analysis in Ethereum threat detection is confirmed by all assessed models having more than 91 percent accuracy in both tasks.

9. Conclusion

The proposed integrated architecture will remove the redundant pre-processing by a common feature pipeline and enable parallel layers of binary and multiclass detection with nine different model configurations. It was revealed that time-interval features computed on 1-day, 3-day, 7-day and 30-day time windows were the most effective features category with a 4.20 percentage point increase in accuracy compared to static features. The framework was tested holistically on the publicly available EtherShield data that can serve as a reproducible experimental benchmark to the blockchain security research community. The main conclusions of this paper are below. First, CNN + BiLSTM hybrid model with the highest binary classification accuracy of 98.890 condition indicates that a combination of local feature extraction by convolutional methods with bi-directional sequencing time-based modelling has a better detection capability. Second, the ISO+RF hybrid achieves the highest multiclass performance of 96.12 % which demonstrates that unsupervised augmentation of the anomaly score significantly increases the performance of the supervised classification, especially where the threat category is rare and complex. Third, Inference latency XGBoost has the highest accuracy-efficiency trade-off with the highest binary accuracy of 98.43 and the lowest inference latency of less than a millisecond, which is why XGBoost is the preferred solution in

deployment situations with constraints in the latency of inference. Fourth, there was over 91% accuracy in all the tested models on both binary and multiclass problems, which confirmed the inherent usefulness of the time-interval behavioral analysis method on Ethereum threat detection. The Ethereum threat detection performance levels of this framework yield a number of state-of-the-art binary accuracy, multiclass accuracy, binary F1-score and macro-averaged multiclass F1-score at 98.89% binary accuracy, 96.12% multiclass accuracy, 98.69% binary F1-score and 96.04% macro-averaged multiclass F1-score on Ethereum threat detection data set, EtherShield. The overall superiority of integrated architectures to standalone models in all metrics of assessment confirms the integrated design philosophy of using anomaly detection together with supervised classification. Per-class analysis establishes good performance of all five categories of threats with minimum F1-scores exceeding 93 showing that the classification of multi-category threats is viable in a practical setting. The suggested framework makes a valuable contribution to the dynamic sphere of blockchain security since it shows that the temporal behavior analysis, hybrid learning models, and unified binary-multiclass design can be used jointly to handle the most challenging problem of automated Ethereum threat detection. With the further rise in blockchain usage rates and the evolution of adversarial threat capabilities, a model of interpretable feature engineering with a strong ensemble and deep-learning classifier will be needed to sustain the security and integrity of decentralized financial infrastructure. The next steps of future development of the framework include its extension to real-time deployment, graph-based analysis, explainable AI integration, and cross-blockchain generalization, which will contribute to the practical influence of the framework in the blockchain threat intelligence and financial fraud prevention ecosystems.

References

1. Alnami, Hani, Muhammad Mohzary, Basem Assiri, and Hussein Zangoti. 2025. "An Integrated Framework for Cryptocurrency Price Forecasting and Anomaly Detection Using Machine Learning" Applied Sciences 15, no. 4: 1864. https://doi.org/10.3390/app15041864
2. Vaddi, L.; Neelisetty, V.; Vallabhaneni, B.C.; Prakash, K.B. Predicting crypto currency prices using machine learning and deep learning techniques. Int. J. 2020, 9, 6603–6608.
3. Kim, H.M.; Bock, G.W.; Lee, G. Predicting Ethereum prices with machine learning based on Blockchain information. Expert Syst. Appl. 2021, 184, 115480.
4. Yuan, Z. Gold and Bitcoin Price Prediction based on KNN, XGBoost and LightGBM Model. Highlights Sci. Eng. Technol. 2023, 39, 720–725.
5. Assiri, B.; Khan, W.Z. Fair and trustworthy: Lock-free enhanced tendermint blockchain algorithm. TELKOMNIKA (Telecommun. Comput. Electron. Control) 2020, 18, 2224–2234.
6. Gao, G., Li, Z., Jin, L., Liu, C., Li, J., & Meng, X. (2025). RTMS: A Smart Contract Vulnerability Detection Method Based on Feature Fusion and Vulnerability Correlations. Electronics, 14(4), 768. https://doi.org/10.3390/electronics14040768
7. Assiri, B.; Khan, W.Z. Enhanced and lock-free tendermint blockchain protocol. In Proceedings of the 2019 IEEE International Conference on Smart Internet of Things (SmartIoT), Tianjin, China, 9–11 August 2019; pp. 220–226.
8. Jumani F, Raza M. Machine Learning for Anomaly Detection in Blockchain: A Critical Analysis, Empirical Validation, and Future Outlook. Computers. 2025; 14(7):247. https://doi.org/10.3390/computers14070247
9. Assiri, B. A modified and effective blockchain model for e-healthcare systems. Appl. Sci. 2023, 13, 12630.
10. Assiri, B. Leader election and blockchain algorithm in cloud environment for e-health. In Proceedings of the 2019 2nd International Conference on New Trends in Computing Sciences (ICTCS), Amman, Jordan, 9–11 October 2019; pp. 1–6.
11. Kuznetsov, O., Kostenko, O., Klymenko, K., Hbur, Z., & Kovalskyi, R. (2025). Machine Learning Analytics for Blockchain-Based Financial Markets: A Confidence-Threshold Framework for Cryptocurrency Price Direction Prediction. Applied Sciences, 15(20), 11145. https://doi.org/10.3390/app152011145
12. Assiri, B.; Alnami, H. Smart Partitioned Blockchain. Sensors 2024, 24, 4033.
13. Alfahaid, A., Alalwany, E., Almars, A. M., Alharbi, F., Atlam, E., & Mahgoub, I. (2025). Machine Learning-Based Security Solutions for IoT Networks: A Comprehensive Survey. Sensors, 25(11), 3341. https://doi.org/10.3390/s25113341
14. Poongodi, M.; Sharma, A.; Vijayakumar, V.; Bhardwaj, V.; Sharma, A.P.; Iqbal, R.; Kumar, R. Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system. Comput. Electr. Eng. 2020, 81, 106527.

15. Mezina A, Ometov A. Detecting Smart Contract Vulnerabilities with Combined Binary and Multiclass Classification. Cryptography. 2023; 7(3):34. https://doi.org/10.3390/cryptography7030034

16. Chen, Z.; Li, C.; Sun, W. Bitcoin price prediction using machine learning: An approach to sample dimension engineering. J. Comput. Appl. Math. 2020, 365, 112395.

17. Kiani, R., & Sheng, V. S. (2024). Ethereum Smart Contract Vulnerability Detection and Machine Learning-Driven Solutions: A Systematic Literature Review. Electronics, 13(12), 2295. https://doi.org/10.3390/electronics13122295

18. Hu, H.; Bai, Q.; Xu, Y. Scsguard: Deep scam detection for ethereum smart contracts. In Proceedings of the IEEE INFOCOM 2022-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Virtual Conference, 2–5 May 2022; IEEE: Piscataway, NJ, USA, 2022; pp. 1–6.

19. Li, P.; Wang, G.; Xing, X.; Zhu, J.; Gu, W.; Zhai, G. A smart contract vulnerability detection method based on deep learning with opcode sequences. Peer-to-Peer Netw. Appl. 2024, 17, 3222–3238.

20. Liu, Y.; Wang, C.; Ma, Y. DL4SC: A novel deep learning-based vulnerability detection framework for smart contracts. Autom. Softw. Eng. 2024, 31, 24.

21. Bani-Hani, R. M., Shatnawi, A. S., & Al-Yahya, L. (2024). Vulnerability Detection and Classification of Ethereum Smart Contracts Using Deep Learning. Future Internet, 16(9), 321. https://doi.org/10.3390/fi16090321

22. Bofeng Pan, Natalia Stakhanova, and Zhongwen Zhu. 2024. EtherShield: Time-interval Analysis for Detection of Malicious Behavior on Ethereum. ACM Trans. Internet Technol. 24, 1, Article 2 (February 2024), 30 pages. https://doi.org/10.1145/3633514

23. Kulkarni, Sanjivani Hemant, et al. "Applied discrete mathematics in developing efficient cryptographic algorithms for enhancing information security in WSN." Journal of Discrete Mathematical Sciences and Cryptography 28.5-A (2025): 1733-142.

24. Joshi, Manjushri, et al. "Building AI-driven frameworks for real-time threat detection and mitigation in IoT networks." 2025 International Conference on Emerging Smart Computing and Informatics (ESCI). IEEE, 2025.