one-quarter of all users and a high portion of Bitcoin transactions are associated with illegal activities [26].

Such activity may be expressed as active accounts of high degrees participating in high-volume transactions.

C. Existing Anomaly Detection Methods and Related Work

Traditional Detection Methods. Some recent surveys focused on security data collection and analysis in different networks [27], [28]. Several related research projects have suggested solutions to detect suspicious accounts: The authors of [6] tried to identify bot activity by inspecting the distribution of time differences between consecutive transactions of Ethereum wallets. In another work, [29] the authors tried to detect smart Ponzi schemes by examining transactions and account features such as textual data that is attached to transactions, account interactions, and transaction frequencies. Statistical study of blockchain transaction flows was undertaken in [7] and [30] to answer interesting questions about the typical behavior of users.

Machine Learning Detection Methods. A considerable amount of the research work that has been done in the field of anomaly detection in blockchain relied on Machine Learning methods to extract the structural features of the blockchain network graph. The blockchain can be represented as a weighted graph describing transactions between accounts [31], [32], [33]. From the graph, features can be extracted using different ML methods such as Graph Convolutional Network (GCN) and autoencoder technology or by using a One-Class SVM in order to detect anomalies. The authors of [34] also applied K-means clustering to detect spam Bitcoin transactions for DoS attacks. The authors of [35] used XGBoost classifier to detect illicit accounts based on their transaction history.

These approaches may yield good results but their point of weakness relies on the fact that in order to identify suspicious activity, one must iterate over all the blocks in the chain and collect all the transactions for further computation.

Frequency Estimation with Sketches. Processing large sequential data streams such as blockchain blocks, network packets or sensors data can be very resource-consuming and time inefficient. Thus, sketching techniques have become widely adopted for various network monitoring and management tasks, including estimating flow size [36], identifying heavy hitters [37], detecting heavy changers [38] and so on. The authors of [39] suggested the usage of two sketches: HyperLogLog [18] and Sample & Hold [19], [40] in order to detect and mitigate Random Subdomain DDoS attacks [41]. The authors of [42] described a method for summarizing large-volume network traffic while allowing reversibility to find anomalous keys. Other notable methods have been proposed to estimate statistics such as cardinality over large volume traffic using sampling algorithms [43], [44], [45] and sketches [46], [47], [48].

In contrast to this paper, none of the mentioned studies attempts to implement frequency estimation on *blockchain networks*. There are some key characteristics of blockchain networks distinguishing them from computer networks which frequency estimation solutions must consider [49]: First, the granularity of transmitted data elements in blockchain are blocks of transactions; whereas in computer networks, packets, which are equivalent to individual transactions in Blockchain, are sent individually. Second, in computer networks, packets are routed from one node to another, while in blockchain, due to it being decentralized, blocks are broadcasted. Lastly, unlike in computer networks, blockchain nodes are anonymous.

Usage of Sketches in Blockchain. The current design of blockchain networks such as Bitcoin and Ethereum makes use of sketches such as Bloom Filter and Merkle Tree to answer membership queries and encode blocks data [2], [3]. Some acclaimed methods proposed using sketches for other purposes as well: [9] surveys potential future applications of sketches in blockchains; [10] suggested using Bloom Filter for account addresses representation in the interest of Bitcoin light clients; and [11] proposed using Invertible Bloom Lookup Table for synchronizing pools of transactions. In contrast to this paper, none of the mentioned studies attempts to use sketches for the purpose of *anomaly detection* in blockchain networks which is challenging in light of the key characteristics of blockchain mentioned previously.

III. Model and Problem Definition

In this article, we focus on the detection of suspicious accounts and suspicious network anomalies. We list particular account and network behaviors that are expressed through characteristics of the transactions. We use $I(\cdot)$ to denote the indicator function.

The first three problems refer to specific activities of suspicious accounts we aim to identify with high accuracy.

Problem 1 (Problem 1: Account Transaction Values): Within a time period, consider the aggregated sum of transaction values in which an account is involved (either as a sender or a receiver). We say that the account is suspicious if this value refers to a major part of the total transactions value at that time. Formally, with some anomaly factor $F \in [0, 1]$, an account A with an aggregated value $TxsValueA$ is said to be suspicious if the total transactions value over all accounts $TxsValueTotal$, satisfies

$$I\big(\text{TxsValueA} \geq F \cdot \text{TxsValueTotal}\big) \tag{1}$$

Problem 2 (Problem 2: Account Transaction Frequency): Within a time period, consider the number of transactions in which an account is involved (either as a sender or a receiver). We say that the account is suspicious if this value refers to a major part of the total number of transactions at that time. Formally, with some anomaly factor $F \in [0, 1]$, an account A with a transaction count $TxsCountA$ is said to be suspicious if the total transactions count over all accounts $TxsCountTotal$, satisfies

$$I\big(\text{TxsCountA} \geq F \cdot \text{TxsCountTotal}\big) \tag{2}$$

Problem 3 (Problem 3: Account Node Degree): Within a time period, consider the node degree of an account namely the number of distinct accounts the account interacts with. We say that the account is suspicious if this value is larger than a specified threshold θ_{deg}.

TABLE I
THREATS ON BLOCKCHAIN THAT CAN BE ENCOUNTERED USING PROBLEMS 1-4

Threat \ Problem	Problem 1	Problem 2	Problem 3	Problem 4
DDoS	×	×		×
Schemes			×	×
Takeover	×	×	×	×
Illicit Activity	×	×	×	×

Formally, with some anomaly degree threshold θ_{deg}, an account A with a node degree $DegreeA$ is said to be suspicious within a time period if it satisfies

$$I\left(\text{DegreeA} \geq \theta_{\text{deg}}\right) \tag{3}$$

Problem 4 (Problem 4: Variety of Estimation Tasks): We aim to detect various traffic anomalies (e.g. identifying heavy hitters, increase detection, entropy estimation etc.) by relying on a single block summary. The motivation for this challenge is that sometimes, the particular phenomena that we aim to detect, might not be known as early as the time of computing the block summaries. Also, we are restricted in size and cannot create a dedicated summary to each potential solution. Hence, we seek to find a multi-task sketch-based solution that offers both *generality* (i.e., supports many applications) but at the same time provides fair *accuracy* comparable to custom solutions (e.g., solutions to Problems 1-3). Such sketch can detect attacks with multiple features, eliminating the need for separate sketches. For example, it can identify DDoS attacks based on changes in both HHs and network transaction entropy, using a single data structure for a more efficient detection process. In addition, a generic sketch enables addressing questions that may arise at a later time after its creation, without requiring prior knowledge of those questions during the sketch's initial creation.

Detecting Threats Using Our Methods. As outlined in Section IIB, threats to the blockchain can be distinguished by certain defining characteristics. Table I illustrates how the problems mentioned above can aid in reducing the risks posed by various threats on the blockchain. Traditionally, in order to measure the statistics described in the table we would have to iterate over the entire blockchain history block by block and read a large number of blocks transactions. This approach is highly unscalable and extremely time and resources consuming.

To allow scalability, we offer a two-stage workflow. In the **online stage**, a sketch is added to the header of each block. The sketches can be computed either by a miner and be part of the block proposal examined in the consensus protocol or alternatively computed by each node independently upon the arrival of the block. Creating a sketch is a relatively simple and quick task compared to the blocks arrival time. In the **offline stage**, sketches of blocks recorded in the blockchain can be processed to detect anomalies efficiently, even upon considering a large number of blocks. This stage can also be referred to as the detection stage.

IV. METHODS FOR SIGNIFICANT ACTIVITY DETECTION

A. Goals, Accuracy Metrics and High-Level Approach

Goals. For Problems 1-3 our goal is to find accounts with a relatively significant aggregated transactions value/frequency or with an absolute significant node degree within a period of time. Our objective for Problem 4 is to introduce a universal framework that can facilitate the resolution of various inquiries.

Accuracy Metrics. Our methods classify accounts as suspicious or non-suspicious based on the sketches. To measure the accuracy of the methods we compare these classifications to those computed based on the complete blocks data. Each classification can be observed as one of four categories: true positive, false positive, true negative or false negative. For instance, false positives refer to accounts classified as suspicious by the sketches and not by the complete data. From these we calculate the metrics Precision $= \frac{TP}{TP+FP}$ and the Recall $= \frac{TP}{TP+FN}$. We also look at the metric $F1\text{-score} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall}+\text{Precision}}$ that considers both Precision and Recall. The purpose of this evaluation method is to increase the chances of sketch-based classification being as effective as full-data based classification in detecting anomalies and real-world attacks.

High-level Approach. Intuitively, for Problems 1-2 our solution stores the top-K accounts with the highest values of transactions or number of transactions in each block. For Problem 3 our approach maintains a HLL data structure that approximates the number of unique neighbors of each account out of the top-K accounts with the highest degree in the block. Finally, for Problem 4 our proposal is to store a generic application-agnostic pyramid sketch for each N blocks window that will be processed offline for a variety of counting tasks.

UTXO (Unspent Transaction Output) and account-based are two different transaction models used in blockchain technology to represent ownership and transfer of value between participants [50]. The main difference between the two models is that in the UTXO model, used by Bitcoin [51], every transaction must refer to previous transactions by using their unspent outputs as inputs to the new transaction. Once a sender is confirmed to have adequate unspent funds, they can submit the transaction and receive the unspent transaction output, which represents the remaining balance. In contrast, the account-based model, used by Ethereum [52], maintains account balances for each participant, similar to a traditional bank account, and represents transactions as debits and credits to these accounts. Our proposed methods are model-agnostic, focusing solely on transaction characteristics such as values, frequencies and degrees, and can be applied to both models.

B. Solution to Problems 1-2: Account Transaction Values and Frequencies

In order to detect accounts with a significant total transactions value or frequency over a period of time we can use a **Heavy Hitters** sketch.

For Problem 1 the sketch includes account addresses of the top-K accounts with the highest aggregated block transactions