Crypto Pump and Dump Detection using Deep Learning Techniques

Ankit Rajvanshi[1] **Ayanesh Chowdhury**[1] **Ujjwal Narain Asthana**[1]
[1]Department of Computer Engineering
[1]New York University
[1]New York, USA
{ar7996,ac10230,una207}@nyu.edu

Abstract

Cryptocurrency fraud detection, despite increased adoption, is inadequately explored, particularly in pump and dump schemes. Existing studies on stock market scams face limitations in applying to cryptocurrencies due to the absence of labeled stock data and distinctive volatility. Current research primarily relies on statistical or classical machine learning models. To address this gap, the proposal suggests using two neural network architectures for pump and dump detection, showing that deep learning outperforms existing methods. This underscores the necessity for sophisticated approaches to combat fraud in the dynamic cryptocurrency landscape.

1 Introduction

In 2021, the trading volume of cryptocurrencies surpassed $14 trillion, marking a staggering 700% increase from the previous year, with Binance contributing to over two-thirds of this total. Despite this rapid growth, concerns about regulatory challenges in the cryptocurrency space, particularly the detection of fraudulent activities, have gained prominence. Compared to traditional financial markets, the crypto industry faces relatively limited regulation, creating an environment where fraud, notably pump and dump (P&D) schemes, can thrive.

P&D schemes are widespread and relatively straightforward to execute in the crypto realm, often orchestrated by a single online planning group. Despite their prevalence, no studies have delved into applying deep learning techniques to detect P&D schemes in cryptocurrencies. While deep learning has been employed in traditional securities like stocks, these studies lack the extensive and freely available data present on the blockchain, and their applicability to the more volatile crypto landscape remains uncertain.

This paper introduces two novel applications of existing deep learning methods designed to identify P&D schemes in small, volatile cryptocurrencies, commonly known as altcoins. The emphasis is on harnessing the wealth of accessible data through deep learning to achieve enhanced performance, as prior research in this field has predominantly relied on classical machine learning techniques or basic statistical analyses.

2 Related Work

2.1 Application of Classical Machine Learning and Statistical Models to Crypto P&D Detection

Until now, the detection of pump and dump (P&D) schemes in the cryptocurrency space has primarily relied on classical machine learning models, specifically random forest trees, as demonstrated in

the work by La Morgia et al. in 2020 [9]. Additionally, statistical models, as applied by Kamps and Kleinberg in 2018 [5], have been employed in this domain. These models typically aggregate trade data at a higher level and utilize these consolidated features to forecast the occurrence of P&D schemes. Given that these approaches constitute the entirety of existing research in this specialized field, our contribution in this paper aims to establish a more robust foundation for future advancements in detecting P&D schemes in the cryptocurrency space.

2.2 Application of Deep Learning to General Anomaly Detection

The realm of anomaly detection, unrelated to cryptocurrencies, is well-explored, with a focus on applying deep learning to time-series anomaly detection problems. Numerous studies have investigated multiple network architectures, including Long Short-Term Memory networks (LSTMs) as demonstrated by Malhotra et al. in 2016 [10], convolutional networks as explored by Kwon et al. in 2018 [8], and various combinations of these architectures, such as those presented by Kim and Cho in 2018 [7]. More recently, attention-based methods, including RNN attention (Brown et al., 2018) [1] and the anomaly transformer (Xu et al., 2021) [13], have also been explored.

As of the writing of this report, deep learning architectures stand at the forefront of time-series anomaly detection. This is attributed to their proficiency in making predictions using spatiotemporal relationships, which are crucial for developing robust anomaly detection models, as highlighted by Kim and Cho in 2018 [7]. In the presented work, the authors implement, modify, and fine-tune some of these cutting-edge architectures with the aim of adapting them to the specific challenges presented by the cryptocurrency domain.

3 Dataset

The dataset utilized in this study comprises manually labeled raw transaction data obtained from the Binance cryptocurrency exchange, as initially introduced by La Morgia et al. in 2020 [9]. The data encompass transactions involving various cryptocurrencies that were identified as experiencing confirmed pump and dump (P&D) incidents.

To construct the dataset, the researchers engaged with several cryptocurrency P&D Telegram groups renowned for planning and executing P&D schemes. Over a two-year period, the team collected timestamps corresponding to official "pump signals" announced by group administrators within these Telegram groups. Leveraging these timestamps and the Binance API, the authors retrieved every transaction associated with the pumped cryptocurrency for a duration of up to one week before and after the pump, depending on accessibility. This process resulted in the collection of data corresponding to 343 P&D occurrences.

Following the retrieval of raw data from the Binance API, the authors conducted further preprocessing by aggregating transactions into 5-second, 15-second, and 25-second "chunks," thereby generating three distinct aggregated datasets. Each of these aggregated datasets consists of 15 features, contributing to a comprehensive understanding of the transactional dynamics surrounding P&D occurrences.

The dataset is characterized by 15 features, providing detailed information related to pump and dump occurrences:

1. **Date, HourSin, HourCos, MinuteSin, MinuteCos:** These features include the date and positional encoding of the hour and minute for a given data chunk.
2. **PumpIndex, Symbol:** These features consist of the 0-based index of the pump, identifying it among the 343 available pumps, and the ticker symbol of the cryptocurrency on which the pump occurred.
3. **StdRushOrder, AvgRushOrder:** These features encompass the moving standard deviation and average percent change in the number of rush orders.
4. **StdTrades:** This feature represents the moving standard deviation of the number of trades, both buy and sell.
5. **StdVolume, AvgVolume:** These features include the moving standard deviation and average percent change in the order volume.

6. **StdPrice, AvgPrice, AvgPriceMax:** These features comprise the moving standard deviation, average percent change, and average maximum percent change in the price of the cryptocurrency asset.

4 Methodology

Since pump and dump (P&D) schemes in the cryptocurrency domain often unfold through multiple phases occurring over significantly different timeframes (e.g., the accumulation phase lasting up to a month, while the pump or dump phases can be as short as a minute), our chosen models possess the capability to capture both longer-term anomalies, referred to as "trend" anomalies, and much shorter-term anomalies, known as "point" anomalies. This dual capability is essential for effectively detecting P&D schemes, as models exclusively focused on one aspect or the other may be susceptible to manipulation by the inherent volatility of crypto markets.

4.1 Proposed Models

We will be applying two well-performing architectures on standard anomaly detection datasets to the growing financial data available in the cryptocurrency space. The selected architectures are the C-LSTM model, as introduced by Kim and Cho in 2018 [7], and the Anomaly Transformer model, developed by Xu et al. in 2021 [13].

4.1.1 C-LSTM

The first model we employ is the C-LSTM model, initially introduced by Kim and Cho in 2018 [7] for learning anomaly detection by treating data as spatiotemporal in nature. This model is structured with a series of convolutional/ReLU/pooling layers to encode the input sequence, succeeded by a set of LSTM layers, and decoding is carried out through a set of feedforward layers. In the C-LSTM model, the convolutional layers play a crucial role in capturing spatial information within the dataset, facilitating the identification of point anomalies. Simultaneously, the LSTM layers contribute to capturing temporal information, aiding in the identification of trend anomalies.

This straightforward model has demonstrated success in detecting various types of web traffic anomalies (Kim and Cho, 2018) [7] and anomalies across multiple stocks in the Chinese stock market (Yang et al., 2020) [4]. The model's effectiveness lies in its ability to handle both spatial and temporal aspects, making it suitable for identifying different anomaly patterns. The visual representation of the C-LSTM model is provided in the figure 1.

4.1.2 Anomaly Transformer

We examined the Anomaly Transformer by Xu et al. (2021) [13] as shown in figure 2, a specialized transformer with an anomaly attention module and minimax optimization for enhanced anomaly detection. Our adaptation includes a unique optimization strategy and loss function for supervised settings, diverging from its original unsupervised design. Chosen for its state-of-the-art performance in time-series anomaly detection, the Anomaly Transformer excels in diverse datasets, such as server sensors (Su et al., 2019) [11], NASA rover data (Keogh et al., 2021) [6], and the NeurIPS 2021 benchmark (Lai et al., 2022) [2].

The Anomaly Transformer's first key innovation for anomaly detection is its unique attention mechanism, replacing traditional self-attention with two internally-calculated values: series and prior associations.

Series association (S_l) at each layer performs simplified self-attention prior to the standard value matrix multiplication, targeting long-term trends to spot trend anomalies.

$$S_l = \textit{Softmax}\left(\frac{QK^T}{\sqrt{d_{\text{model}}}}\right)$$

Here: - S_l is the output of the softmax function. - Q represents the query matrix. - K represents the key matrix. - T denotes the transpose operation. - p is the positional encoding. - d_{model} is the dimensionality of the model.

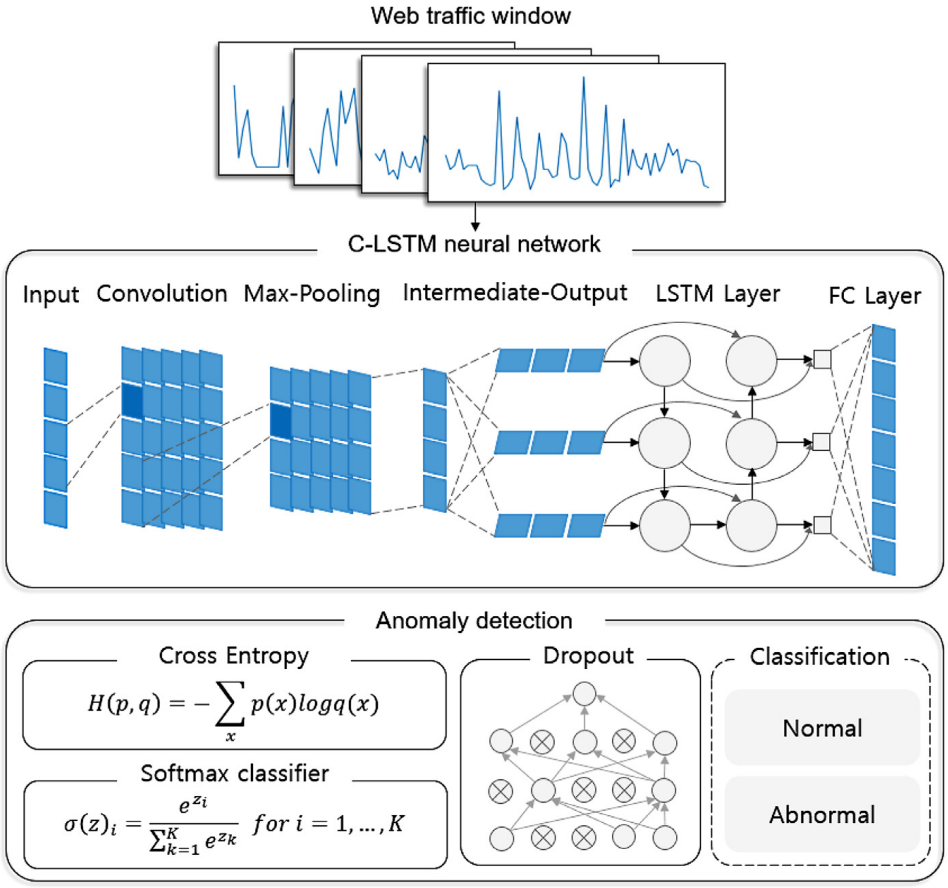

Figure 1: C-LSTM Architecture [7]

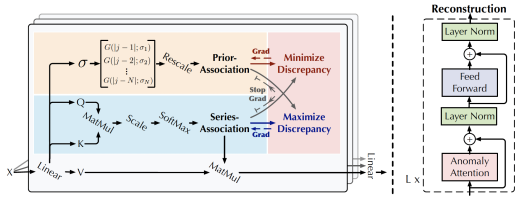

Figure 2: Anomaly Transformer [13]

Prior association (P_l) employs a Gaussian kernel on each input point, converting results into discrete distributions at each layer, honing in on local data for point anomaly detection.

$$P_l = Rescale([\mathcal{N}(j|\mu = i, \sigma)]_{i,j\epsilon 1,\ldots,N})$$

The second key innovation by this model is a two-phase minimax optimization as shown in figure 3 : the minimize phase, where the series association aligns with the prior association using symmetric KL divergence (SKL), and the maximize phase, where it aligns with the input sequence, emphasizing non-adjacent points and expanding via SKL. This process lowers attention to anomalies in a sequence generated by both associations. Differences between associations are quantified by the Association Discrepancy (AD) function. Post-optimization, data reconstruction involves multiplying the series association (S_l) with the standard value matrix (V) of an attention module.

The reconstructed series X', when anomalies are present, will significantly differ from the original, enhancing discrepancy and aiding anomaly detection.

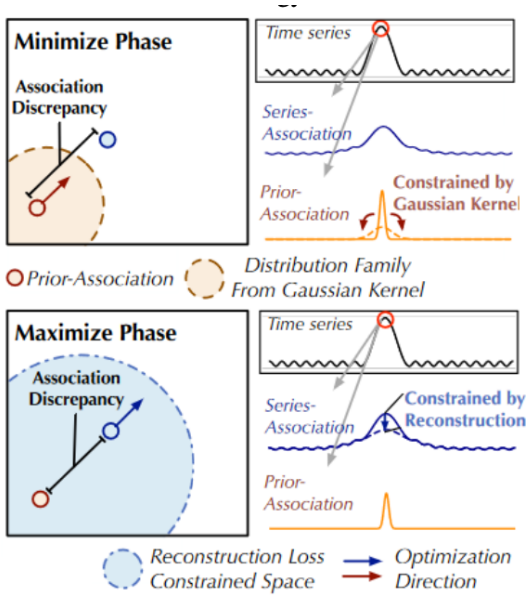

Figure 3: Minimax Strategy [13]

4.2 Performance Metrics

During our experiments, we evaluate the performance of our models on the validation set at the end of each epoch by collecting precision, recall, and F1 scores. In the context of cryptocurrency pump and dump (P&D) detection, which is essentially an anomaly detection problem, we choose not to include accuracy in our evaluation metrics. This decision is based on the recognition that accuracy measurements may not accurately reflect the strength of a model in anomaly detection scenarios.

Anomaly detection scenarios often exhibit a large class imbalance, heavily favoring negative labels. In such cases, the accuracy of any model tends to be exceptionally high, even if it simply outputs the majority class (negative class). As a result, accuracy is not a reliable indicator of a model's performance in anomaly detection. Instead, we optimize our models for the best possible F1 score. F1 scores are particularly suitable for anomaly detection models, as they provide a balanced assessment of precision and recall. This standard aligns with the approach adopted by most anomaly detection models and facilitates a more effective comparison of our results with previous work.

The precision is defined as the percentage of predicted anomalies that were correctly classified, while the recall is defined as the percentage of actual anomalies that were correctly classified as anomalies. The F1 score is the harmonic mean of these two values. Given that both precision and recall are crucial in the context of anomaly detection, the F1 score, being a balanced combination of the two, is a robust metric for comparing different models.

The following equations are used to calculate precision, recall, and the F1 score, respectively:

$$\text{Precision} = \frac{NTP}{NTP + NFP}$$

$$\text{Recall} = \frac{NTP}{NTP + NFN}$$

$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Where:
NTP represents the number of true positives (correctly predicted anomalies).
NFP represents the number of false positives (actual non-anomalies predicted as anomalies).

NFN represents the number of false negatives (actual anomalies predicted as non-anomalies).

4.3 Implementation

In the subsection, we mention more specific details about our technique to detect pump and dump schemes.

4.3.1 Data Pre-processing

In the initial step, given an input data sequence $X = X_1, \ldots, X_N$, the data is divided into separate training and validation sets using an 80:20 ratio without shuffling. After this split, the training data undergoes the following preprocessing steps:

1. The data is segmented into M contiguous subsequences, denoted as $X = Y_1, \ldots, Y_M$, where Y_i corresponds to the i-th pump in the dataset determined by the PumpIndex feature. This segmentation ensures that, during training, the models are not exposed to information from different pumps simultaneously, preventing potential interference with the training process.
2. Pump sequences Y_i with fewer than 100 chunks are excluded from the training subset. This exclusion is based on the rationale that pumps with a limited number of chunks may not serve as optimal training samples. Consequently, the training subset consists of $m < M$ pump sequences meeting this criterion.

Following the initial segmentation of pump sequences, each pump Y_i undergoes additional data preparation steps:

1. **Segmentation into Windows:** Each pump Y_i is split into segments of size s by applying a sliding window over the chunks of the pump. To ensure consistency in the number of windows, reflection padding of size $s-1$ is added to the start of each pump. This adjustment ensures that the resulting count of windows matches the original count of chunks. Within this scheme, the chunks within each window are collectively considered a single "segment."
2. **Model Input and Prediction:** Segments serve as inputs to the models, which then predict the probability of a pump occurring during the last chunk of the segment.
3. **Shuffling and Batching:** After this stage, segments from all pumps can be safely shuffled and batched together. This process mitigates the risk of information from one pump "leaking" into the training data for another pump, ensuring that the models are trained effectively.

The decision to segment the data is motivated by several considerations. Firstly, this segmentation and prediction approach mitigates the risk of models making predictions based on future values. In other words, in this setup, models can only predict whether or not a pump will occur based on currently-known information. This ensures that models trained under this scheme possess the capability to be deployed in real-world, real-time anomaly detection scenarios on actual exchanges.

Moreover, the choice to segment the data is influenced by the architectural characteristics of the C-LSTM model, which includes an LSTM layer. LSTM layers may encounter performance limitations for longer sequence lengths (as noted by Vaswani et al. in 2017) [12]. Since the data for each pump can span multiple days, training models directly on entire pump sequences may pose challenges due to memory constraints. Having fixed segment lengths, as opposed to variable lengths of pump data, simplifies the data loading process. This segmentation strategy facilitates practical deployment and addresses architectural constraints associated with the chosen model.

The decision to segment our data into chunks offers a straightforward approach for performing undersampling, a technique proven to enhance model performance in anomaly detection scenarios by addressing class imbalance. This improvement is observed even when the reduction does not bring the class ratio all the way down to 50:50, as highlighted in studies such as Hasanin and Khoshgoftaar (2018) [3]. Undersampling not only improves model performance but also naturally accelerates training by reducing the number of training samples.

In our approach, we implement undersampling on the training data by selectively retaining only a random subset, proportional to u, of all segments that do not contain any anomaly labels. Simultane-

ously, we unconditionally include all segments that have an anomaly label anywhere in the dataset. This strategy optimizes training efficiency and addresses class imbalance to enhance the overall effectiveness of the anomaly detection model.

The validation data undergoes a similar segmenting process, but with a few exceptions. Unlike the training data, small pumps are not discarded, and no undersampling is applied. This decision is made to prevent unfairly skewing metrics in favor of our models during the validation process. By preserving small pumps and refraining from undersampling, the validation set remains representative of the entire dataset, allowing for a more unbiased evaluation of model performance.

4.3.2 C-LSTM Implementation

Our custom C-LSTM model features the following architecture: one set of convolutional/ReLU/pooling layers with a convolution kernel size of 3 and a stride of 1, along with a pooling kernel size of 2 and a stride of 1. Additionally, it incorporates one LSTM layer with an embedding dimension of 350, one feedforward layer that directly projects the last hidden state of the LSTM to a dimension of 1, and a sigmoid layer that confines the output of our classifier between 0 and 1.

This configuration yields a C-LSTM model with 997,851 learnable parameters, and the model undergoes training for 200 epochs to optimize its performance.

4.3.3 Anomaly Transformer Implementation

Our supervised adaptation of the Anomaly Transformer involved modifying its optimization strategy and loss function, originally designed for unsupervised data. We made these key changes for supervised anomaly detection:

1. Retained the maximize phase but altered the minimize phase to align series association with ground truth.
2. Substituted the initial loss term with MSE loss, comparing output and ground truth labels.

The revised maximize phase loss function and our model configuration – a 15 sequence length, 4 layers, and a 0.0001 lambda – led to a model with 3,030 learnable parameters, trained over 50 epochs.

5 Results and Conclusion

We benchmark our models against the outcomes of the random forest model utilized in the study conducted by (La Morgia et al., 2020) [9].

Model	Dataset	Precision	Recall	F1
C-LSTM	**5S**	**88.70%**	**78.3%**	**83.2%**
C-LSTM	**15S**	**91.7%**	**90.2%**	**91.0%**
C-LSTM	25S	91.7%	88.7%	90.2%
Anomaly Trans.	5S	80.00%	66.7%	72.7%
Anomaly Trans.	15S	84.9%	73.8%	78.9%
Anomaly Trans.	25S	96.1%	79.0%	86.7%
RF[9]	5S	92.20%	77.50%	82.70%
RF[9]	15S	91.10%	83.30%	87.00%
RF[9]	25S	93.10%	91.40%	92.00%

Github Repository Link: Crypto Pump and Dump Detection using Deep Learning Techniques

Traditional ML models like Random Forest usually outperform deep learning models when the dataset is small. But we can see that C-LSTM model outperforms the Random Forest benchmark for the 5 second and the 15 second datasets. The reason for this is that unlike random forest, C-LSTM models are designed to capture temporal dependencies in sequential data (LSTM layers are present). Also, convolutional layers in the C-LSTM model can automatically learn hierarchical features from the input data. This feature extraction capability is beneficial when the dataset has complex spatial or temporal patterns that may be challenging for Random Forest to capture effectively.

Anomaly Transformer models unfortunately does not do that. A significant reason for that is the lack of data. In the case of Anomaly Transformer, after undersampling we are left with 22,743, 26,837, 36,215 entries in the 5 second, 15 second and 25 seconds dataset respectively, which is a very small number to train attention-based models.

We also found that predictions using the 5-second chunked dataset are much less accurate than those on the 15-second and 25-second chunked dataset, which suggests that predicting anomalies using smaller chunk sizes corresponds to a harder problem in general. This corroborates findings from previous works (La Morgia et al., 2020) [9].

In summary, while the C-LSTM model excels in certain scenarios, the choice of the most suitable model may depend on the specific requirements of the application, such as the desired trade-off between precision and recall. But it certainly serves as a strong contender when it comes to capturing spatio-temporal relationships as demonstrated in this project.

References

[1] Andy Brown, Aaron Tuor, Brian Hutchinson, and Nicole Nichols. Recurrent neural network attention mechanisms for interpretable system log anomaly detection. In *Proceedings of the first workshop on machine learning for computing systems*, pages 1–8, 2018.

[2] Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. Adbench: Anomaly detection benchmark. *Advances in Neural Information Processing Systems*, 35:32142–32159, 2022.

[3] Tawfiq Hasanin and Taghi Khoshgoftaar. The effects of random undersampling with simulated class imbalance for big data. In *2018 IEEE international conference on information reuse and integration (IRI)*, pages 70–79. IEEE, 2018.

[4] Chuangxia Huang, Xian Zhao, Renli Su, Xiaoguang Yang, and Xin Yang. Dynamic network topology and market performance: A case of the chinese stock market. *International Journal of Finance & Economics*, 27(2):1962–1978, 2022.

[5] Josh Kamps and Bennett Kleinberg. To the moon: defining and detecting cryptocurrency pump-and-dumps. *Crime Science*, 7(1):1–18, 2018.

[6] Dutta Roy T. Naik U. Agrawal A Keogh, E. Multi-dataset time-series anomaly detection competition, sigkdd 2021. `https://compete.hexagon-ml.com/practice/competition/39/`, 2021.

[7] Tae-Young Kim and Sung-Bae Cho. Web traffic anomaly detection using c-lstm neural networks. *Expert Systems with Applications*, 106:66–76, 2018.

[8] Donghwoon Kwon, Kathiravan Natarajan, Sang C Suh, Hyunjoo Kim, and Jinoh Kim. An empirical study on network anomaly detection using convolutional neural networks. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 1595–1598. IEEE, 2018.

[9] Massimo La Morgia, Alessandro Mei, Francesco Sassi, and Julinda Stefa. Pump and dumps in the bitcoin era: Real time detection of cryptocurrency market manipulations. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–9. IEEE, 2020.

[10] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

[11] Chenglin Miao, Wenjun Jiang, Lu Su, Yaliang Li, Suxin Guo, Zhan Qin, Houping Xiao, Jing Gao, and Kui Ren. Privacy-preserving truth discovery in crowd sensing systems. *ACM Transactions on Sensor Networks (TOSN)*, 15(1):1–32, 2019.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[13] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Anomaly transformer: Time series anomaly detection with association discrepancy. *arXiv preprint arXiv:2110.02642*, 2021.