mathematics

MDPI

Article

Ensemble Approach Using k-Partitioned Isolation Forests for the Detection of Stock Market Manipulation

Hugo Núñez Delafuente [1,*], César A. Astudillo [2,*] and David Díaz [3]

[1] Doctorado en Sistemas de Ingeniería, Faculty of Engineering, Universidad de Talca, Curicó 3340000, Chile
[2] Department of Computer Science, Faculty of Engineering, Universidad de Talca, Curicó 3340000, Chile
[3] Departamento de Administración, Facultad de Economía y Negocios, Universidad de Chile, Santiago 8330111, Chile; ddiaz@fen.uchile.cl
* Correspondence: hnunez@utalca.cl (H.N.D.); castudillo@utalca.cl (C.A.A.)

Abstract: Stock market manipulation, defined as any attempt to artificially influence stock prices, poses significant challenges by causing financial losses and eroding investor trust. The prevalent reliance on supervised learning models for detecting such manipulations, while showing promise, faces notable hurdles due to the dearth of labeled data and the inability to recognize novel manipulation tactics beyond those explicitly labeled. This study ventures into addressing these gaps by proposing a novel detection framework aimed at identifying suspicious hourly manipulation blocks through an unsupervised learning approach, thereby circumventing the limitations of data labeling and enhancing the adaptability to emerging manipulation strategies. Our methodology involves the innovative creation of features reflecting the behavior of stocks across various time windows followed by the segmentation of the dataset into k subsets. This setup facilitates the identification of potential manipulation instances via a voting ensemble composed of k isolation forest models, which have been chosen for their efficiency in pinpointing anomalies and their linear computational complexity—attributes that are critical for analyzing vast datasets. Evaluated against eight real stocks known to have undergone manipulation, our approach demonstrated a remarkable capability to identify up to 89% of manipulated blocks, thus significantly outperforming previous methods that do not utilize a voting ensemble. This finding not only surpasses the detection rates reported in prior studies but also underscores the enhanced robustness and adaptability of our unsupervised model in uncovering varied manipulation schemes. Through this research, we contribute to the field by offering a scalable and efficient unsupervised learning strategy for stock manipulation detection, thereby marking a substantial advancement over traditional supervised methods and paving the way for more resilient financial markets.

Keywords: stock market manipulation; unsupervised learning; voting ensemble; anomaly detection; isolation forest

MSC: 68T01

check for updates

Citation: Núñez Delafuente, H.; Astudillo, C.A.; Díaz, D. Ensemble Approach Using k-Partitioned Isolation Forests for the Detection of Stock Market Manipulation. *Mathematics* **2024**, *12*, 1336. https://doi.org/10.3390/math12091336

Academic Editors: Luca Di Persio, Matteo Garbelli and Anatoliy Swishchuk

Received: 17 March 2024
Revised: 9 April 2024
Accepted: 23 April 2024
Published: 27 April 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Stock market manipulation constitutes a deliberate attempt to distort the genuine prices of assets, thereby misleading investors and affecting their investment decisions. This manipulation not only triggers economic losses for investors but also compels the state to divert its limited resources towards monitoring and controlling such activities. Furthermore, companies targeted by manipulation efforts experience significant reputational damage, thus compounding the negative impact beyond mere financial loss [1]. The U.S. Securities and Exchange Commission (SEC) succinctly defines stock market manipulation as intentional actions aimed at deceiving or defrauding investors through the control or artificial influence of asset prices. The study of stock market manipulation is crucial for

maintaining market integrity, protecting investor interests, and ensuring the smooth functioning of financial markets. Understanding and combating such manipulative practices are essential for fostering a transparent, fair, and efficient market environment where investors can make decisions based on accurate and truthful information.

There is a prolific research stream seeking to understand the manipulation process [2,3]. For instance, Allen and Gale [4] classified manipulation activities according to how they are performed into three categories: action-based, rumor-based, and trading. They also showed that an uninformed manipulator can benefit by mimicking the behavior of an informed trader with the help of information asymmetries. The work of the International Organization of Securities Commissions [5] describes what methods manipulators use, with the main ones being wash sales, advancing the bid, pumping and dumping, marking the close, cornering the market, and squeezing the market. The work of Imisiker et al. [6] analyzes the characteristics of manipulated shares, thereby concluding that companies that were previously manipulated and have high leverage ratios also have a high probability of being manipulated, while stocks with a high volume available for trading and high market capitalization are difficult to manipulate.

However, few studies seek to detect and predict manipulation, and even fewer use machine learning tools for detection [2,7]. The generally strong performance of supervised learning models can largely be attributed to their ability to learn from patterns explicitly presented to them. By employing labeled data, these models are trained to recognize and respond to the specific patterns they have been exposed to during the training process. This focused learning approach, however, introduces a significant limitation: the model's difficulty in identifying novel manipulation patterns—those which it has not been previously taught [8]. This inherent challenge emphasizes the need for models that can adapt to and detect emerging patterns of manipulation, thus extending beyond the confines of their initial training set.

Palshikar et al. [9] performed one of the first investigations that allowed for detecting manipulation using fuzzy temporal logic; they identified the common trading patterns used by manipulators. Öğüt et al. [2] used probabilistic neural networks (PNNs) and support vector machines (SVMs) to obtain better results in detecting manipulation cases than those obtained with traditional statistical models.

Diaz et al. [10] used an unsupervised approach to identify manipulated hourly blocks, which were then used as labels for a supervised analysis. Using decision trees, they extracted different rules to identify manipulation patterns.

Cao et al. [11] introduced a novel semisupervised learning methodology that employs a hidden Markov model (HMM) specifically designed for the task at hand. This approach, dubbed the Hidden Markov Model with Abnormal States (HMMAS), was strategically applied to analyze stock data from both the NASDAQ and London Stock Exchange, thereby aiming to uncover patterns indicative of market manipulation. In developing HMMAS, the authors posited certain assumptions about the underlying data distribution, thus creating a solid foundation for the targeted detection of manipulation within these major financial markets.

Yang et al. [12] conducted a comparative analysis of various supervised learning algorithms with the objective of identifying suspected cases of market manipulation. Among the algorithms evaluated, the naive Bayes classifier emerged as the most effective, thus demonstrating superior performance in detecting potential manipulation instances.

Leangarun et al. [13] used long short-term memory generative adversarial networks (LSTM-GANs) to achieve 68.1% accuracy when identifying manipulated cases. Wang et al. [7] combined the characteristics derived from commercial records and those of listed companies and used recurrent neural networks (RNNs) to detect manipulation activities. Their results were, on average, 29.8% higher in terms of area under the ROC curve (AUC) than those observed in studies that used traditional statistical tools. Rizvi et al. [14] proposed an unsupervised model based on the idea of learning the relationship between stock prices in the form of an affinity matrix; the characteristics extracted from this matrix were used

to train an autoencoder. Finally, they used clustering based on kernel density estimation (MKDE) to detect manipulated operations, where nonclustered data were treated as manipulated. Rizvi et al. [8] used kernel PCA to obtain vectors of characteristics delivered to MKDE to detect manipulations. To this end, they used a dataset with information on 13 stocks from NASDAQ and the London Stock Exchange (LSE), with the information of manipulations generated in synthetic form. Leangarun et al. [15] compared the LSTM autoencoder (LSTM-AE) and LSTM-GANs, with both models identifying five of six manipulations and yielding a low false positive rate. Models based on deep learning show promising results. However, they are limited by high computational complexity [16,17].

From the studies reviewed, it is evident that employing a supervised learning approach carries the inherent risk of overfitting, where the algorithm might become excessively tailored to the labeled manipulation patterns at hand, thereby compromising its ability to generalize to new or unseen data. This risk is particularly pronounced in fields like stock market manipulation detection, where labeled data are scarce, thus making it crucial to mitigate overfitting to maintain model robustness. To address this challenge, our proposal advocates for the exploration of unsupervised learning techniques, which, by not relying on labeled data, naturally avoid the pitfalls of overfitting and potentially offer a more generalized and adaptable solution. Furthermore, a critical examination of the reported success rates and the transparency of false positive results, as emphasized by Rizvi et al. [8], are essential steps to validate the effectiveness of these unsupervised approaches in real-world applications.

This study aims to detect manipulation activity using an unsupervised learning approach. To bolster the detection capabilities for anomalies, the dataset was augmented with new features derived from sophisticated statistical calculations. We performed manipulation detection using a voting ensemble composed of unsupervised anomaly detection models. We employed a real dataset to evaluate the performance of our proposal; this consists of annual data from eight stocks that have undergone manipulation activities.

This article centers on the transformative impact of the Isolation Forest (IF) algorithm in detecting stock market manipulation through an unsupervised learning approach. The isolation forest, distinguished by its innovative use of isolation rather than density or distance to identify anomalies, offers a unique advantage in the financial domain where manipulative activities are often subtle and masked within vast datasets. The unsupervised nature of this algorithm eliminates the need for a prelabeled dataset, thus addressing the challenge of scarce labeled data in the realm of financial fraud detection. Moreover, its efficiency in handling high-dimensional data and its scalability make it particularly suitable for the dynamic and complex environment of the stock market [18–20]. By deploying this method, our study sheds light on its efficacy in uncovering manipulation patterns, thereby contributing to safer and more transparent financial markets.

The main contributions of this research are (1) proposing an unsupervised manipulation detection strategy that improves the task of identifying manipulated time blocks and (2) presenting the benefits of using a voting ensemble approach to detect manipulated blocks.

The remainder of this paper is organized as follows. Section 2 describes the methodology used, the case study, the description of the voting ensemble model and the Isolation Forest algorithm, the performance measures used to evaluate the model, and ends with details of the model implementation. Section 3 presents the results of the model in the search of manipulations; these are compared with the results of previous studies. Section 4 discusses the results, in addition to making recommendations and observing weaknesses. Finally, Section 5 summarizes the results and proposes future research directions.

2. Materials and Methods

This section details the methodology used to identify suspected manipulation cases using an unsupervised approach.

2.1. Methodology

Adopting the Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology in our investigation into the capabilities of an isolation forest ensemble for detecting stock market manipulation offers a structured, iterative, and comprehensive framework that significantly enhances the study's scientific rigor and practical applicability. The CRISP-DM methodology has been previously and successfully employed in other machine learning projects, as evidenced by the literature [21–23]. By meticulously following CRISP-DM's phases—from understanding the business problem and data to model evaluation and deployment—we ensure a deep alignment between our models and the real-world phenomenon of market manipulation. This approach not only guarantees the transparency and repeatability of our experiment but also ensures that our findings are directly applicable to real-world scenarios. The iterative nature of CRISP-DM allows for continuous model refinement, thus leading to optimized detection capabilities. Furthermore, the methodology's emphasis on understanding business objectives and data intricacies ensures that our ensemble models are both effective in anomaly detection and relevant to the specific challenges of stock market manipulation, thereby providing a clear pathway for deploying these models in practical trading systems. Figure 1 illustrates the distinct phases of the CRISP-DM methodology as implemented in our study.

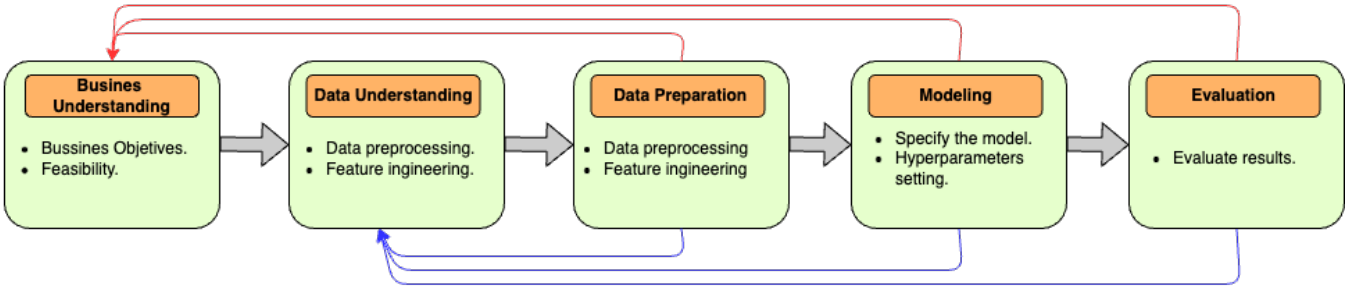

Figure 1. CRISP-DM methodology stages used.

2.2. Case Study

The dataset analyzed includes cases of manipulation identified during 2003 and pursued through litigation actions by the U.S. Securities and Exchange Commission (SEC). These data were used by [10,24] and consist of 12,748 instances with time information for the January–December period. Eight stocks were affected by manipulation activities during the analyzed period. There is certainty that these stocks were manipulated during 2003, but the total number of affected transactions is not known. In [10], the authors manually labeled 55 cases containing manipulated transactions, which were used to evaluate the model performance. Table 1 shows the number of manipulations per stock.

Table 1. Manipulations per stock.

Stock	Number of Manipulated Blocks
AKSY	10
BIF	7
BTF	7
ESPR	13
FF	5
JHFT	6
OME	1
ZAP	6
TOTAL	55

The 55 labeled manipulated transactions were identified by reviewing lawsuits filed by the SEC. Manipulated stocks are those that include words related to market manipulation in the lawsuits filed by the SEC, for example, "manipulation" and "marking the close", or those referring to Sections 9(a) or 10(b) of the Securities and Exchange Act (1934), which are articles relating to market manipulation.

Initially, we selected variables commonly used in stock market analysis. These variables are price, return, volume, and number of transactions. Figure 2 shows some selected stocks' temporal distribution of manipulated blocks.

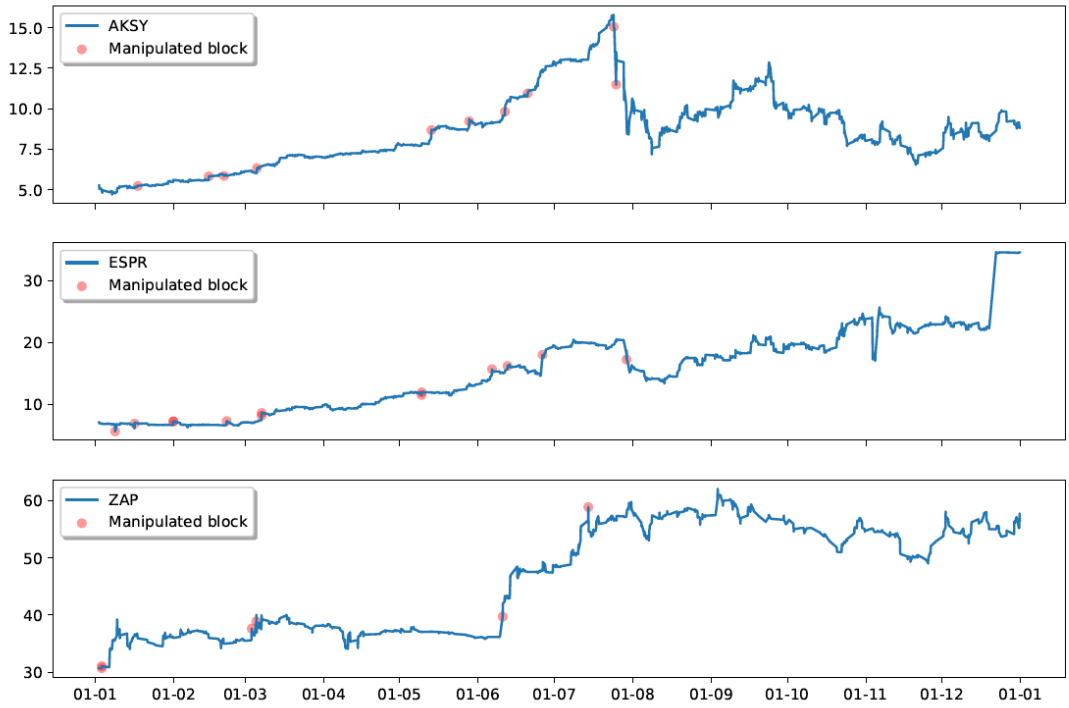

Figure 2. Price and manipulated blocks of AKSY, ESPR, and ZAP.

2.3. Ensemble Approach Using k-Partitioned Isolation Forests

In our investigation, we focus on harnessing the power of unsupervised learning to unearth fraudulent activities, thus utilizing the Isolation Forest algorithm across k distinct partitions of the dataset. In this ensemble strategy and as shown in Figure 3, the dataset is partitioned into distinct subsets by dividing it along its columns into k random partitions. For each subset, the Isolation Forest algorithm is applied independently. The final determination of whether manipulation has occurred is made through a majority voting process among the outcomes of the Isolation Forest applications across all subsets. The assumption is that the ensemble approach enhances the robustness of fraud detection by aggregating multiple independent assessments, thus potentially capturing a broader range of manipulative activities within the dataset.

This study leverages the unsupervised learning capabilities of the Isolation Forest algorithm, thus deploying it across k unique partitions of our dataset to detect fraudulent activities. By segmenting the dataset into k random subsets, each defined by a division of the dataset's columns, we apply the Isolation Forest algorithm to each subset independently. The collective judgement on the occurrence of manipulation is then derived from a majority vote across the results from these distinct isolation forest applications. This ensemble method is predicated on the notion that combining insights from multiple, independently assessed partitions increases the detection accuracy by capturing a wider spectrum of potential fraudulent behaviors. The dataset underpinning our analysis contains four fundamental variables central to stock market analytics, as recognized by prior research [25–27]: price, return, volume, and trade count. A feature engineering process

augments these variables with additional metrics that reveal the temporal dynamics of the market more comprehensively. This includes integrating moving averages to mitigate transient fluctuations, volatility indices to gauge price movements, calculations of abnormal returns to spotlight outliers, and standardizing these metrics into z scores and ratios for uniform assessment. After refining our dataset with these additional metrics, the features are randomly allocated into k separate sets. This division paves the way for generating k data subsets, with each offering a distinct lens for the anomaly detection task. Such an ensemble framework ensures that each instance is evaluated from multiple angles, with its classification as either anomalous or normal determined by the consensus from all analyses. The threshold for deeming an instance as manipulative is clearly established in Equation (1), thus facilitating a detailed and precise mechanism for spotting manipulations.

$$\sum_{i=1}^{k} v_i \geq \text{threshold} \tag{1}$$

In the described ensemble approach, k represents the total number of classifiers deployed within each ensemble, with each classifier tasked with analyzing a specific partition of the data. The variable v_i denotes the vote cast by the anomaly detector for partition i, thereby adopting a binary format where a vote of 1 signifies the classification of the instance as anomalous, and a vote of 0 indicates a normal classification. The decision threshold, a critical parameter in this setup, determines the minimum number of votes an instance must receive to be deemed manipulated. This threshold, along with the specific k values utilized in our study, are detailed in Table 2. This mechanism allows for a nuanced aggregation of classifier decisions, thereby ensuring that an instance is only classified as manipulated if it surpasses the predefined threshold of consensus among the ensemble's classifiers, thereby enhancing the precision and reliability of the detection process.

Table 2. Thresholds to be evaluated for each k value used.

k	Threshold
1	{1}
2	{1, 2}
3	{1, 2, 3}

Furthermore, we juxtaposed the outcomes from this ensemble strategy against results derived from applying the anomaly detection algorithm directly to the unpartitioned, original dataset. This comparison underscores the efficacy of the k-partitioned ensemble approach in enhancing fraud detection capabilities.

2.4. Performance Metrics

We evaluated the performance in terms of recall, precision, F1 Score (F1), and F2 Score (F2), which are commonly used metrics in this type of problem [28–31].

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$F1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} \tag{4}$$

$$F2 = \frac{5}{\frac{4}{\text{recall}} + \frac{1}{\text{precision}}} \tag{5}$$

Precision in Equation (2) is the ratio of correctly detected manipulations over the total manipulations identified by the model. In Equation (3), the recall corresponds to

the proportion of correctly identified manipulations out of the total number of manipulations. True Positives (TPs) represent the number of manipulations correctly identified by the model, False Positives (FPs) correspond to the number of nonmanipulated cases incorrectly identified as manipulated, and False Negatives (FNs) show the number of manipulated cases that are incorrectly classified as nonmanipulated. The F1 score, as defined in Equation (4), serves as the harmonic mean between precision and recall, thus ensuring that both metrics contribute equally to the overall score. This balanced approach makes the F1 score particularly useful for scenarios where an even emphasis on precision and recall is desired. Conversely, the F2 score, outlined in Equation (5), adjusts this balance by diminishing the weight of precision while amplifying that of recall. This modification is especially relevant in contexts where the cost of false negatives is higher than that of false positives, thereby making recall a more critical measure. For both the F1 and F2 scores, the optimal achievable value is 1, thereby indicating perfect precision and recall, while the least desirable score is 0, thus signifying the lowest performance in these metrics.

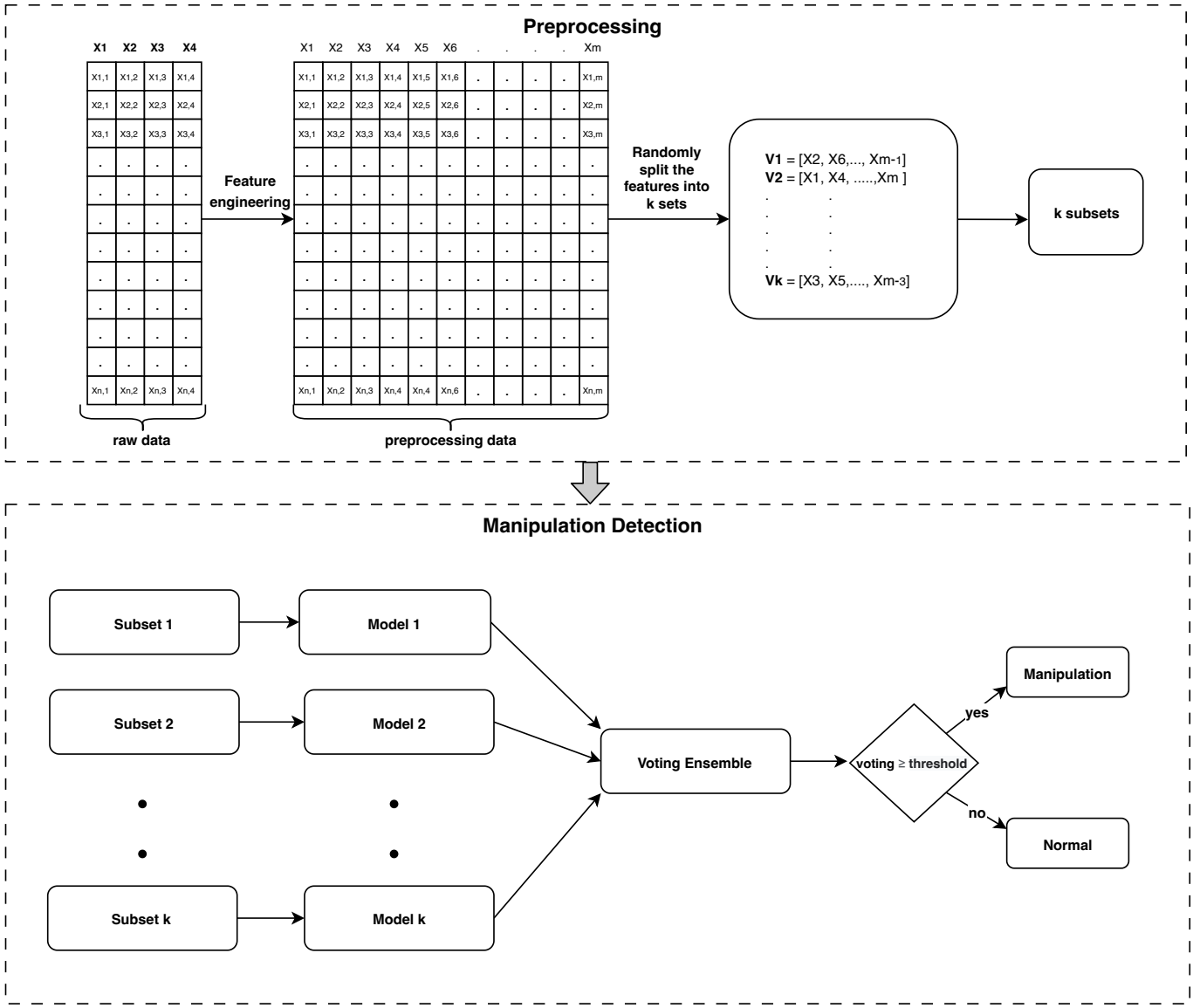

Figure 3. Representation of manipulations detection strategy.

2.5. Anomaly Detection Algorithm

We employ Isolation Forest (IF) [32] as the anomaly detection algorithm. IF is an unsupervised algorithm based on decision trees. The main idea behind using IF is that

anomalous instances can be isolated from normal ones through the recursive partitioning of the dataset. This algorithm has been successfully used in different application fields, for example, to detect credit card fraud [33] and health insurance fraud [34], for software and UAV failure prediction [35,36], and in detecting unusual water consumption [37]. Mendes et al. [38] observed that IF outperformed more complex models in detecting anomalies.

Algorithm 1, known as Isolation Forest, is a novel approach specifically tailored for anomaly detection within datasets. This algorithm diverges from traditional methods by exploiting the inherent properties of anomalies being 'few and different', thereby isolating them efficiently. At its core, the Isolation Forest algorithm utilizes a collection of Isolation Trees (iTrees), as described in Algorithm 2, to partition the data. Each iTree is constructed by recursively selecting a feature at random and then choosing a split value between the maximum and minimum values of the selected feature until instances are isolated or a predefined depth limit is reached. The crux of assessing an observation's anomaly score lies in the PathLength method outlined in Algorithm 3. This method calculates the length of the path traversed in an iTree to isolate a sample, thus serving as a proxy for its anomaly score. Shorter paths indicate a higher likelihood of being anomalies, as they are easier to isolate. By averaging the path lengths over a forest of iTrees, the Isolation Forest algorithm provides a robust measure of an observation's deviation from the norm, thus enabling effective and efficient anomaly detection in large datasets.

Algorithm 1: *iForest*

Input:
X input data.
t—number of trees.
ψ—subsampling size.
Ouput: a set of t *iTrees*

Initialize *Forest*
set height limit $l = ceiling(log_2\ \psi)$
for $i = 1$ **to** t **do**
    $X' \leftarrow sample(X'\ \psi)$
    $Forest \leftarrow Forest \cup iTree(X', 0, \mathrm{l})$
end for
return *Forest*

Algorithm 2: *iTree(X,e,l)*

Input:
X input data.
e - current tree height.
l height limit.
Ouput: an *iTree*

if $e \geq l$ *or* $|\ X\ | \leq 1$ **then**
    return *exNode*{*Size*← |\ X\ |\ }
else
    let Q be a list of attributes in X
    randomly select an attribute q $\in$ Q
    randomly select a split point p from *max* and *min* values of attribute q in X
    $X_l \leftarrow$ *filter*(X, q $<$ p)
    $X_r \leftarrow$ *filter*(X, q $\leq$ p)
    return *inNode* {Left←*iTree*(X_l,e+1,l),
    Right←*iTree*(X_l,*e*+1,*l*),
    SplitAtt←*q*,
    SplitValue←*p* }
end if

Algorithm 3: *PathLength* (x, T, e)

Input:
x - an instance.
T - an iTree.
e - current path length; to be initialized to zero when first called
Ouput: path length of x

if T *is an external node* **then**
  return $e + c(T.size)$
end if
$\alpha \leftarrow T.splitAtt$
if $x_\alpha < T.splitAtt$ **then**
  return $Pathlength\ (x, T.left, e + 1)$
else
  $\{X_a \geq T.splitValue\}$
  return $Pathlength\ (x, T.right, e + 1)$
end if

Equation (6) presents the formula for calculating the anomaly score [32], denoted as $S(x, n)$, for an observation x within a dataset by employing the Isolation Forest algorithm. This formula is crucial for assessing the anomaly degree of an instance in relation to the rest of the dataset. The equation is defined as

$$S(x, n) = 2^{-\frac{E(h(x))}{c(n)}} \tag{6}$$

where:

$S(x, n)$ represents the anomaly score of the observation x in a dataset of size n.

$E(h(x))$ signifies the average path length (calculated by the PathLength method) from the root to the terminal node across all instances of Isolation Trees (iTrees) in the forest. This value reflects how quickly the observation x can be isolated in the iTrees forest.

$c(n)$ is a normalization constant that depends on the dataset size n, thus ensuring that the score is independent of the dataset's size and remains within a comparable range.

The factor $2^{-\frac{E(h(x))}{c(n)}}$ normalizes the outcome so that scores fall within a range of 0 to 1, where values close to 1 indicate a high likelihood of being an anomaly, while values closer to 0 suggest the observation is normal.

The Isolation Forest algorithm requires the specification of two critical hyperparameters for its operation: the subsample size (ψ) and the number of trees (t). As advised by the creators of the algorithm [32], the recommended default values for these hyperparameters are set at 256 for the subsample size ($\psi = 256$) and 100 for the number of trees ($t = 100$). These default settings were empirically determined to provide a balance between computational efficiency and the algorithm's effectiveness in isolating anomalies within a dataset.

2.6. Implementation

In this study, the experiments were meticulously carried out using Python. To identify anomalies within the dataset, we leveraged the Isolation Forest library (https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html, accessed on 7 November 2023), an integral component of the scikit-learn toolkit. The selection of hyperparameter values was in strict accordance with the recommendations provided in [32], thus ensuring an optimal configuration of the Isolation Forest algorithm for our specific use case. We defined an instance as anomalous if its anomaly score S exceeded the threshold of 0.5, thereby allowing us to precisely target and investigate instances most likely indicative of manipulation within our analysis framework.

3. Results

The results of this research are divided into two stages: (1) constitutes preprocessing, which includes the feature engineering phase and the creation of k voting sets, and (2) corresponds to manipulation detection of the 55 confirmed cases, which Isolation Forest performs by voting.

3.1. Preprocessing

We created 27 new features representing moving averages for different time windows, ratios, z scores, variability, and preconditions corresponding to specific days and time zones requiring special attention. Table 3 shows some examples of these new features; the description of all features used in this paper is detailed in Table A1.

Table 3. Examples of created features. In the description, 20 and 30 indicate the number of periods used for the calculation.

Feature	Description
X5	Abnormal Return
X6	Volume Moving Average—30
X7	Trade Moving Average—30
X8	Return Volatility—30
X11	Return Z score—30
X14	Trade Z score—30
X17	Return/Moving Average Ratio—20
X20	Trade Moving Average Rate—20

The delineation of features across individual subsets adheres to a columnar partitioning scheme, facilitated by the parameter k, which is crucial for segmenting the dataset into distinct parts. In a more formal context, given a dataset comprising N features (columns) and a specified partitioning parameter k, the dataset undergoes a division into k subsets along its columns. This process entails distributing the total number of columns N across k subsets, thereby ensuring an equitable distribution of features while maintaining the integrity of each row's data points. This columnwise partitioning method is fundamental for conducting a thorough and segmented anomaly detection analysis, thereby allowing for the dataset's intrinsic properties to be analyzed in a compartmentalized yet comprehensive manner. Table 4 shows detailed information on the number of features per subset as a function of the different values of k used. V_i represents the list of features used to construct each subset. For $k = 2$, the subsets comprise 16 and 15 features.

Table 4. Features per subset.

k	Number of Features
1	$V_1 = 31$
2	$V_1 = 16$, $V_2 = 15$
3	$V_1 = 10$, $V_2 = 10$, $V_3 = 11$

We randomly distributed features to each subset if $k > 1$. To evaluate the possible influence of the assignment of features to subsets on the results and, at the same time, to validate the performance obtained by our manipulated detection strategy, we repeated the experiment 100 times when $k > 1$.

3.2. Detection of Stock Market Manipulation

3.2.1. Detection of Manipulation Using a Single Isolation Forest

In these experiments, we opted for $k = 1$, meaning that anomaly detection was entrusted to a single model within the ensemble. Consequently, if the Isolation Forest

algorithm classifies a specific time block as anomalous, that block is then labeled as manipulated. This approach simplifies the decision-making process by placing the entire burden of anomaly identification on a singular model, thereby directly correlating the detection of anomalies with instances of manipulation within the dataset.

The manipulation detection process on the raw data predicted 1.627 suspected cases of manipulation, of which 36 correspond to confirmed cases. Using the preprocessed set composed of 31 features, the number of suspected cases decreased to 1.255, of which 38 correspond to confirmed cases. Table 5 shows the results obtained.

The inclusion of additional features into the dataset resulted in a notable enhancement of performance metrics, thus achieving a precision score of 0.03. This represents an improvement when compared to a previous study by [10], which processed the same dataset and reported a lower precision of 0.019, while achieving the same level of recall. This enhancement underscores the value of feature engineering in boosting the model's ability to accurately identify instances of interest, thereby refining the overall efficacy of the anomaly detection process.

Table 5. Results obtained using raw data and the preprocessed dataset.

Input	Recall	Precision	F1	F2
Raw data	0.655	0.022	0.043	0.097
Preprocessed data	0.691	0.030	0.058	0.129

3.2.2. Detection of Market Manipulation through Ensemble Methods

To implement this strategy, the preprocessed dataset underwent partitioning into k subsets, which was achieved through the random selection of columns to ensure variability in the analysis. This partitioning is pivotal in our ensemble approach, which utilizes multiple isolation forests to enhance detection capabilities. Specifically, we explored the efficacy of this method with $k = 2$ and $k = 3$ to assess how varying degrees of partitioning impact performance. The decision threshold, critical for determining the criteria under which an instance is classified as manipulated, is meticulously outlined in Table 2. This ensemble method, leveraging the collective insights of several isolation forests, aims to significantly improve the precision and reliability of identifying manipulative activities within the dataset.

As a first experiment, we divided the preprocessed set into two subsets (k = 2), and the manipulation threshold was set to 1. On average, this strategy predicted 982 suspected manipulation cases, of which 44 are real manipulated blocks. Table 6 shows the results obtained in this experiment. The highest number of detected manipulated cases was 49. This was observed in 5 out of the 100 tests.

Table 6. Results when the preprocessed dataset is split into two voting sets (k = 2), and the manipulation threshold = 1.

Metric	Mean	Max	Min	std
Recall	0.800	0.891	0.691	0.069
Precision	0.045	0.055	0.033	0.004
F1	0.084	0.104	0.064	0.008
F2	0.018	0.220	0.141	0.016

We observed that the minimum values of the metrics in Table 6 are equal to or higher than those presented in Table 5. Adopting the voting ensemble strategy, with a value of k equal to 2 and a threshold of 1, is a better option for detecting manipulations than the direct application of the Isolation Forest algorithm on the raw or preprocessed dataset.

Figure 4 presents the average outcomes of our experiments, thereby highlighting that the optimal precision was achieved by applying the maximum threshold for each specified value of k. Furthermore, a comprehensive assessment through the F score, which

considers both recall and precision, reveals that both the F1 and F2 scores consistently excelled within the ensemble setups. Additionally, our analysis uncovers that, irrespective of the selected *k* value, employing enesembles (the green bars) with a minimum threshold of 1 vote consistently yielded superior average results compared to the single classifier options (the blue bars). This observation underscores the efficacy of integrating a voting ensemble approach in enhancing the overall detection performance (Files S1 and S2).

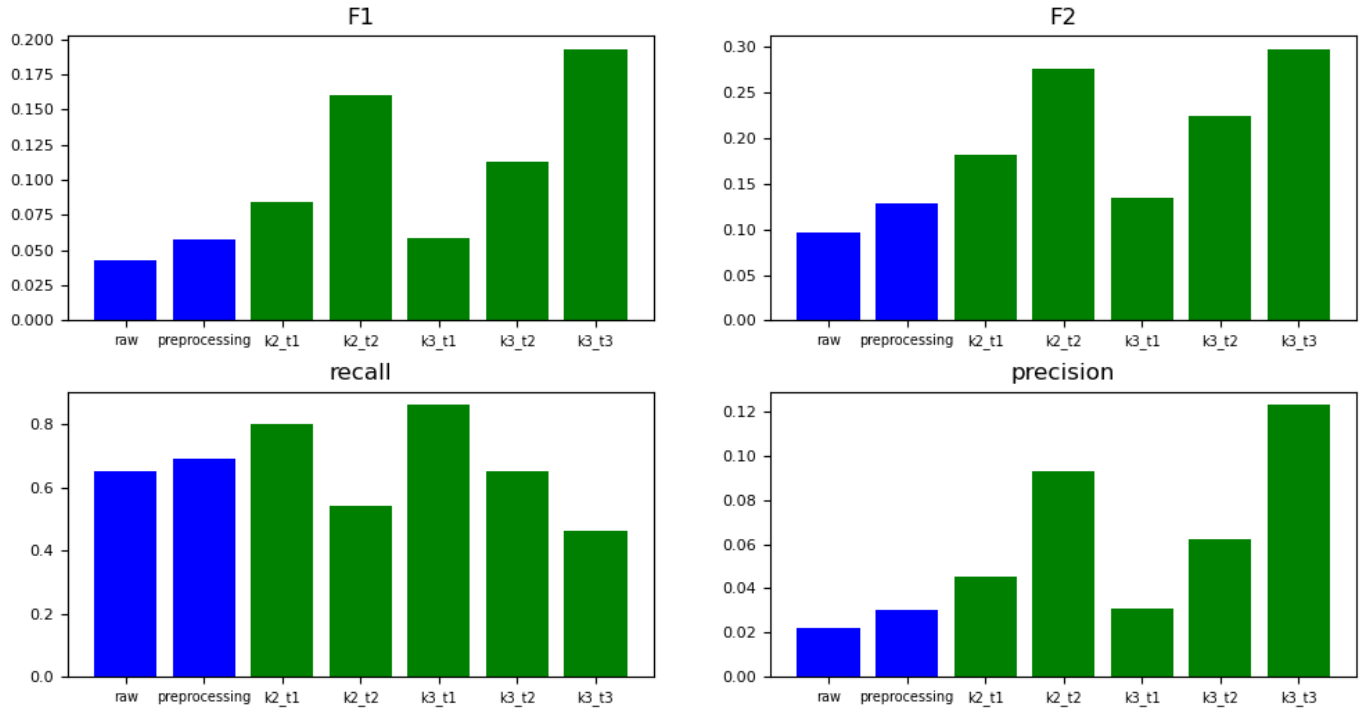

Figure 4. The average results of the different experiments performed. The numbers in the experiment name represent the subsets and the threshold used. For example, k2_t1 indicates that the dataset is divided into two subsets, and the threshold is set to 1.

4. Discussion

The implementation of an unsupervised method for detecting stock manipulation, such as the Isolation Forest algorithm, offers significant advantages over supervised methods. Firstly, the unsupervised approach eliminates the need for previously labeling large datasets, which is a process that can be both costly and prone to errors, especially in dynamic and complex contexts like financial markets. Moreover, unsupervised methods are particularly skilled at identifying anomalies or atypical patterns without prior knowledge, which is crucial for uncovering new forms of stock manipulation that have not yet been documented. Another benefit is their ability to adapt and evolve with real-time data, thus providing more agile and accurate detections in an environment that is constantly changing. This adaptability contrasts with supervised systems, which may require frequent retraining and manual adjustments to maintain their effectiveness against new manipulation tactics. Therefore, an unsupervised approach not only offers a more efficient and less labor-intensive solution for fraud detection but also excels in its capacity to preempt emerging fraudulent strategies, thus strengthening the integrity of the stock market.

In the original dataset used in this research, each stock was represented through four distinct time series, thus posing a unique challenge when applying Isolation Forest for fraud detection. This algorithm, primarily designed for static datasets, necessitated an adaptation to handle time series data effectively. By converting the dynamic nature of time series into a static dataset enriched with statistical features across multiple columns, we were able to imbue the Isolation Forest algorithm with the ability to comprehend historical patterns within various transactions. This transformation is pivotal for a couple of reasons.

Firstly, it addresses the inherent limitation of Isolation Forest in processing sequential data, which constitute a common characteristic of financial transactions. By aggregating time series into a set of descriptive statistics, we preserve essential temporal characteristics without compromising the algorithm's integrity. Secondly, this approach allows for a more nuanced detection of anomalies. Traditional fraud detection mechanisms might struggle to differentiate between naturally occurring fluctuations and genuine instances of fraud. The enriched dataset provides a multidimensional view of each transaction, thus highlighting anomalies that would otherwise remain obscured in raw time series data.

In the ensemble method employing k isolation forests within our study, each classifier is trained on randomly selected columns, thereby forming k distinct partitions of the dataset. This design intentionally positions each isolation forest as a "weak classifier", given that the random column selection limits the scope of data each classifier is exposed to. This limitation is strategic, as it diversifies the analytical perspectives across the ensemble, albeit at the cost of individual classifier robustness. Despite their designation as weak classifiers, the strength of the ensemble approach emerges from aggregating these diverse, partially informed classifiers through a majority voting mechanism. This integration of decisions from across the ensemble capitalizes on the varied insights each weak classifier contributes, based on its unique subset of features. Consequently, this method enhances the overall anomaly detection capability, thereby effectively compensating for the inherent limitations of individual classifiers. The ensemble's collective intelligence, derived from amalgamating the outputs of multiple weak classifiers, significantly boosts the precision and reliability of stock market manipulation detection, thus demonstrating the efficacy of this approach in navigating complex datasets with nuanced patterns of fraud.

In the discussion of our findings, it is crucial to highlight the remarkable improvements facilitated by the adoption of a voting ensemble strategy. Our empirical analysis demonstrates that simply selecting two random feature sets significantly enhances performance metrics. Specifically, with $k = 2$ and a voting threshold of 1, we observed a substantial uplift in the effectiveness of our approach: recall improved by an average of 14.3%, while precision saw a remarkable increase of 46.7% in comparison to the baseline performance of a singular classifier model. Moreover, when analyzing the top-performing configurations within our experiments, the enhancements become even more pronounced. The most effective ensemble setups yielded increases as notable as 28.9% in recall and an impressive 83.3% in precision. These findings underscore the potent capability of the voting ensemble strategy not just to outstrip the performance of individual classifiers, but to do so with considerable margins, thereby reinforcing the value of ensemble methods in complex anomaly detection scenarios such as stock market manipulation.

In the realm of fraud detection, the significance of recall is notably magnified, as highlighted in the literature [39]. This emphasis stems from the understanding that the consequences of overlooking a genuine case of fraud carry far more weight than mistakenly flagging a legitimate transaction as suspicious. Within the context of our voting strategy, it was observed that for each k value implemented, the optimal recall rate was achieved when the voting threshold was set to 1. Our experiments have meticulously explored scenarios with k values of one, two, and three—relatively modest numbers. This naturally raises the intriguing question of the effects that an increased number of classifiers might have on performance metrics and, crucially, on determining the optimal voting threshold. While this line of inquiry is undoubtedly of interest to researchers and holds the potential to further refine fraud detection methodologies, it extends beyond the scope of our current study. Nonetheless, it underscores a promising avenue for future research, thereby inviting a deeper exploration into the scalability of the voting ensemble strategy and its implications for enhancing the detection of financial fraud.

Figure 5 elucidates the nuanced relationship between the threshold and various performance metrics: while recall demonstrated an inverse correlation with the threshold, precision, F1 score, and F2 score exhibited a direct correlation. This dynamic can be comprehended by observing that an increase in the threshold leads to a reduction in both true

manipulated cases (TP) and suspected cases (TP + FP), with a more pronounced decrease in the latter. This trend primarily stems from a significant reduction in False Positives (FPs), as detailed in Table 7. Consequently, the impact on recall was relatively modest compared to the more pronounced sensitivity of precision to threshold adjustments. In essence, precision exhibits a greater responsiveness to changes in the threshold compared to recall, thus highlighting the intricate balance between these metrics in optimizing fraud detection performance.

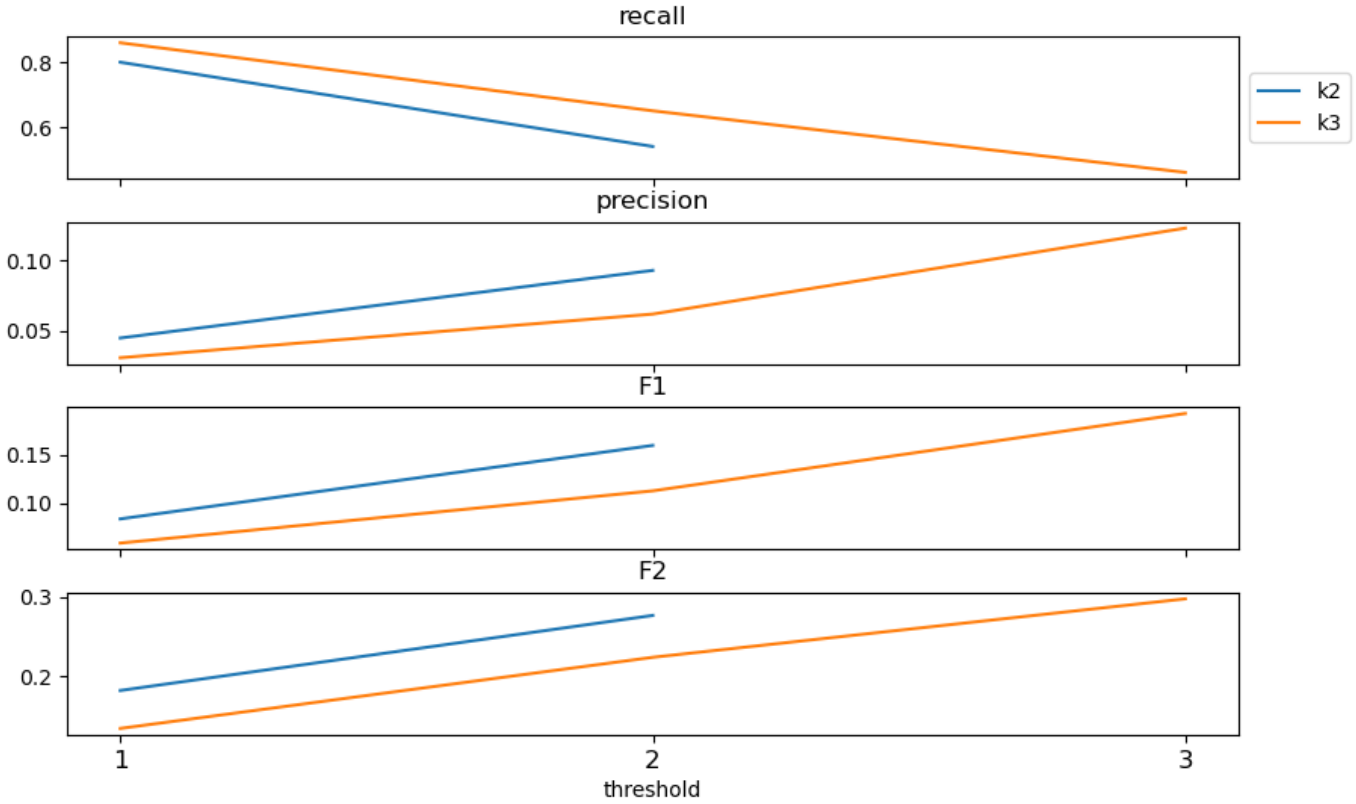

Figure 5. Evaluation of the performance obtained using different thresholds.

Table 7. Changes in TP, FP, and TP + FP as the threshold is increased ($k = 3$). Values are expressed as a percentage of the result obtained with a threshold of 1 vote.

Threshold	TP	FP	TP + FP
2	−24.4%	−63.8%	−62.6%
3	−46.3%	−87.7%	−86.4%

In our analysis, a crucial point of consideration is the inherent uncertainty surrounding the exact number of manipulated blocks within the dataset. This ambiguity introduces a potential for false positives—instances identified by our model as manipulations which do not match known cases of manipulation. However, it is important to acknowledge that these so-called false positives might, in reality, represent genuine instances of manipulation that have not been previously identified or documented. This scenario underscores a limitation in our validation process, where the benchmark for model accuracy is constrained by the completeness and reliability of the manipulation cases available for comparison.

Given this context, the presence of false positives in our results does not necessarily denote model inaccuracy but rather highlights the potential for our methodology to uncover new and unrecorded manipulations. This possibility emphasizes the dynamic and complex nature of stock market manipulation detection, where the discovery of new manipulation patterns can enhance the overall performance of the detection model.

5. Conclusions

This research aimed to detect stock market manipulation using an unsupervised approach. To this end, we proposed a voting ensemble strategy composed of k unsupervised anomaly detection models and evaluated the above on eight real datasets of stocks affected by manipulation activities. To assess the voting ensemble strategy's performance, we used the ability to identify 55 manipulated time blocks.

To enhance the precision of our anomaly detection efforts, we engineered new features that facilitated the creation of data subsets. These subsets were then subjected to a collective decision-making process by the anomaly detection models, thus utilizing the Isolation Forest algorithm as our primary tool for identifying anomalies. An instance was classified as manipulated based on whether it garnered votes surpassing a predetermined threshold. Our findings compellingly demonstrate that the application of a voting ensemble strategy markedly boosted all measured performance metrics, thus surpassing outcomes reported in prior research. Remarkably, a mere division of the dataset into two subsets for voting, coupled with a threshold set to one, was sufficient to elevate performance indicators significantly. Notably, an increase in the voting threshold was found to substantially enhance precision, thus reducing the number of cases flagged for further investigation and, consequently, diminishing the resource expenditure required for audits. By employing this strategic voting mechanism, we achieved an identification of up to 89% of genuinely manipulated blocks, thus underscoring the profound potential of our approach in contributing to the integrity and surveillance of financial markets.

For future work, we aim to extend our investigation by assessing the effectiveness of the voting ensemble strategy in conjunction with an anomaly detection approach that focuses on the reconstruction error of time series. This exploration will delve into how discrepancies in reconstructed time series data can serve as a robust indicator of anomalies, as well as how the incorporation of a voting mechanism may further refine and improve the detection process. This direction promises to offer valuable insights into the nuanced dynamics of time series anomaly detection and the potential synergies with ensemble methodologies.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/math12091336/s1, File S1: average results; File S2: Box plots.

Author Contributions: Conceptualization, H.N.D.; Data curation, H.N.D. and D.D.; Formal analysis, H.N.D., C.A.A. and D.D.; Funding acquisition, C.A.A.; Investigation, H.N.D. and C.A.A.; Methodology, H.N.D.; Project administration, H.N.D.; Supervision, C.A.A.; Validation, H.N.D. and C.A.A.; Writing—original draft, H.N.D.; Writing—review and editing, H.N.D., C.A.A. and D.D. All authors have read and agreed to the published version of the manuscript.

Funding: Hugo Nuñez Delafuente received funding support from the Chilean National Agency of Research and Development, ANID, scholarship grant program: PFCHA/Doctorado Becas Chile/2021—21211244.

Data Availability Statement: The data presented in this study may be available on reasonable request from the first author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Description of the variables employed, with values denoting the number of periods incorporated into their respective calculations.

Feature	Description
X1	Volume
X2	Trade
X3	Price

Table A1. *Cont.*

Feature	Description
X4	Return
X5	Abnormal Return
X6	Volume Moving Average—30
X7	Trade Moving Average—30
X8	Return Volatility—30
X9	Return to Riskmetrics Volatility
X10	Abnormal Return to RiskMetrics
X11	Return Z score—30
X12	Z score de retorno normal—30
X13	Z score de volumen transado—30
X14	Trade Z score—30
X15	Return Volatility–Moving Average Volatility Rate—30
X16	Return to Risk Metrics Volatility–Moving Average Volatility Rate—30
X17	Return/Moving Average Ratio—20
X18	Abnormal return–Moving Average Rate—20
X19	Volume Moving Average Rate—20
X20	Trade Moving Average Rate—20
X21	Price Moving Average Rate—30
X22	Return Volatility Moving Average Rate—30
X23	Abnormal Return Moving Average Rate—30
X24	Volume Moving Average Rate—30
X25	Trade Moving Average Rate—30
X26	Return to Risk Metrics Z score—30
X27	Abnormal Return to Risk Metrics Z score—30
X28	Volatility Z score—30
X29	Return Risk Metrics Z score—30
X30	Suspicious Dates Indicator
X31	Preconditions

References

1. Hanke, M.; Hauser, F. On the effects of stock spam e-mails. *J. Financ. Mark.* **2008**, *11*, 57–83. [CrossRef]
2. Öğüt, H.; Mete Doğanay, M.; Aktaş, R. Detecting stock-price manipulation in an emerging market: The case of Turkey. *Expert Syst. Appl.* **2009**, *36*, 11944–11949. [CrossRef]
3. Zhai, J.; Cao, Y.; Ding, X.; Cao, Y.; Ding, X. Data analytic approach for manipulation detection in stock market. *Rev. Quant. Financ. Account.* **2018**, *50*, 897–932. [CrossRef]
4. Allen, F.; Gale, D. Stock-Price Manipulation. *Rev. Financ. Stud.* **1992**, *5*, 503–529. [CrossRef]
5. International Organization of Securities Commissions, Technical Committee. *Investigating and Prosecuting Market Manipulation*; Technical Committee; International Organization of Securities Commissions: Madrid, Spain, 2000; pp. 1–101.
6. Imisiker, S.; Kamil, B.; Tas, O. Which firms are more prone to stock market manipulation? *Emerg. Mark. Rev.* **2013**, *16*, 119–130. [CrossRef]
7. Wang, Q.; Xu, W.; Huang, X.; Yang, K. Enhancing intraday stock price manipulation detection by leveraging recurrent neural networks with ensemble learning. *Neurocomputing* **2019**, *347*, 46–58. [CrossRef]
8. Rizvi, B.; Belatreche, A.; Bouridane, A.; Watson, I. Detection of Stock Price Manipulation Using Kernel Based Principal Component Analysis and Multivariate Density Estimation. *IEEE Access* **2020**, *8*, 135989–136003. [CrossRef]
9. Palshikar, G.; Bahulkar, A.; Keshav Palshikar, G. Fuzzy Temporal Patterns for Analyzing Stock Market Databases. In Proceedings of the International Conference on Advances in Data Management, Dallas, TX, USA, 15–18 May 2000; Tata-McGraw Hill: Pune, India, 2000; pp. 135–142.
10. Diaz, D.; Theodoulidis, B.; Sampaio, P. Analysis of stock market manipulations using knowledge discovery techniques applied to intraday trade prices. *Expert Syst. Appl.* **2011**, *38*, 12757–12771. [CrossRef]
11. Cao, Y.; Li, Y.; Coleman, S.; Belatreche, A.; McGinnity, T.M. A hidden markov model with abnormal states for detecting stock price manipulation. In Proceedings of the Proceedings—2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013, Manchester, UK, 13–16 October 2013; pp. 3014–3019. [CrossRef]
12. Yang, F.; Yang, H.; Yang, M. Discrimination of China's stock price manipulation based on primary component analysis. In Proceedings of the 2014 International Conference on Behavioral, Economic, and Socio-Cultural Computing (BESC2014), Shanghai, China, 30 October–1 November 2014; pp. 1–5. [CrossRef]

13. Leangarun, T.; Tangamchit, P.; Thajchayapong, S. Stock price manipulation detection using generative adversarial networks. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 2104–2111.
14. Rizvi, B.; Belatreche, A.; Bouridane, A.; Mistry, K. Stock Price Manipulation Detection based on Autoencoder Learning of Stock Trades Affinity. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8. [CrossRef]
15. Leangarun, T.; Tangamchit, P.; Thajchayapong, S. Stock Price Manipulation Detection Using Deep Unsupervised Learning: The Case of Thailand. *IEEE Access* **2021**, *9*, 106824–106838. [CrossRef]
16. Maji, P.; Mullins, R. On the reduction of computational complexity of deep convolutional neural networks. *Entropy* **2018**, *20*, 305. [CrossRef]
17. Schmitt, M. Deep learning in business analytics: A clash of expectations and reality. *Int. J. Inf. Manag. Data Insights* **2023**, *3*, 100146.
18. Guo, Y.; Jiang, X.; Tao, L.; Meng, L.; Dai, C.; Long, X.; Wan, F.; Zhang, Y.; Van Dijk, J.; Aarts, R.M.; et al. Epileptic seizure detection by cascading isolation forest-based anomaly screening and EasyEnsemble. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**, *30*, 915–924. [CrossRef]
19. Shao, C.; Du, X.; Yu, J.; Chen, J. Cluster-based improved isolation forest. *Entropy* **2022**, *24*, 611. [CrossRef] [PubMed]
20. Wei, S.; Yang, D.; Zhang, W.; Zhang, S. A novel noise-adapted two-layer ensemble model for credit scoring based on backflow learning. *IEEE Access* **2019**, *7*, 99217–99230. [CrossRef]
21. Kharitonov, A.; Nahhas, A.; Pohl, M.; Turowski, K. Comparative analysis of machine learning models for anomaly detection in manufacturing. *Procedia Comput. Sci.* **2022**, *200*, 1288–1297. [CrossRef]
22. Pahuja, L.; Kamal, A. EnLEFD-DM: Ensemble Learning Based Ethereum Fraud Detection Using CRISP-DM Framework. *Expert Syst.* **2023**, *40*, e13379. [CrossRef]
23. Silva-Aravena, F.; Núñez Delafuente, H.; Gutiérrez-Bahamondes, J.H.; Morales, J. A hybrid algorithm of ML and XAI to prevent breast cancer: A strategy to support decision making. *Cancers* **2023**, *15*, 2443. [CrossRef]
24. Golmohammadi, K.; Zaiane, O.R.; Diaz, D. Detecting stock market manipulation using supervised learning algorithms. In Proceedings of the DSAA 2014—2014 IEEE International Conference on Data Science and Advanced Analytics, Shanghai, China, 30 October–1 November 2014; pp. 435–441. [CrossRef]
25. Pan, W.T.; Qian, W.B.; He, Y.; Wang, Z.X.; Liu, W. Research on Identifying Stock Manipulation using GARCH Model. *Int. J. Adv. Comput. Sci. Appl.* **2023**, *14*, 956–967. [CrossRef]
26. Ergün, H.O.; Yalaman, A.; Manahov, V.; Zhang, H. Stock market manipulation in an emerging market of Turkey: How do market participants select stocks for manipulation? *Appl. Econ. Lett.* **2021**, *28*, 354–358. [CrossRef]
27. Leangarun, T.; Tangamchit, P.; Thajchayapong, S. Stock price manipulation detection using a computational neural network model. In Proceedings of the 2016 Eighth International Conference on Advanced Computational Intelligence (ICACI), Chiang Mai, Thailand, 14–16 February 2016; pp. 337–341. [CrossRef]
28. Ruchay, A.; Feldman, E.; Cherbadzhi, D.; Sokolov, A. The Imbalanced Classification of Fraudulent Bank Transactions Using Machine Learning. *Mathematics* **2023**, *11*, 2862. [CrossRef]
29. Silva-Aravena, F.; Delafuente, H.N.; Astudillo, C.A. A Novel Strategy to Classify Chronic Patients at Risk: A Hybrid Machine Learning Approach. *Mathematics* **2022**, *10*, 3053. [CrossRef]
30. Alwadain, A.; Ali, R.F.; Muneer, A. Estimating Financial Fraud through Transaction-Level Features and Machine Learning. *Mathematics* **2023**, *11*, 1184. [CrossRef]
31. Yu, K.; Shi, W.; Santoro, N. Designing a Streaming Algorithm for Outlier Detection in Data Mining—An Incrementa Approach. *Sensors* **2020**, *20*, 1261. [CrossRef] [PubMed]
32. Liu, F.T.; Ting, K.M.; Zhou, Z.H. Isolation Forest. In Proceedings of the Eighth IEEE International Conference on Data Mining, IEEE (2008), Pisa, Italy, 15–19 December 2008; pp. 413–422.
33. Kumar, P.; Iqbal, F. Credit Card Fraud Identification Using Machine Learning Approaches. In Proceedings of the 2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT), Chennai, India, 25–26 April 2019; pp. 1–4. [CrossRef]
34. Bauder, R.A.; Da Rosa, R.C.; Khoshgoftaar, T.M. Identifying medicare provider fraud with unsupervised machine learning. In Proceedings of the 2018 IEEE 19th International Conference on Information Reuse and Integration for Data Science, IRI 2018, Salt Lake City, UT, USA, 6–9 July 2018; pp. 285–292. [CrossRef]
35. Ding, Z.; Xing, L. Improved software defect prediction using Pruned Histogram-based isolation forest. *Reliab. Eng. Syst. Saf.* **2020**, *204*, 107170. [CrossRef]
36. Khan, S.; Liew, C.F.; Yairi, T.; McWilliam, R. Unsupervised anomaly detection in unmanned aerial vehicles. *Appl. Soft Comput.* **2019**, *83*, 105650. [CrossRef]
37. Nofal, S.; Alfarrarjeh, A.; Abu Jabal, A. A use case of anomaly detection for identifying unusual water consumption in Jordan. *Water Supply* **2021**, *22*, 1131–1140. [CrossRef]

38. Mendes, T.; Cardoso, P.J.; Monteiro, J.; Raposo, J. Anomaly Detection of Consumption in Hotel Units: A Case Study Comparing Isolation Forest and Variational Autoencoder Algorithms. *Appl. Sci.* **2022**, *13*, 314. [CrossRef]
39. Chung, J.; Lee, K. Credit Card Fraud Detection: An Improved Strategy for High Recall Using KNN, LDA, and Linear Regression. *Sensors* **2023**, *23*, 7788. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.