Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

SECURE Framework for Detecting Integrity Attacks in High-Frequency Trading Systems

S Malathy

malathy10112001@gmail.com

Dwaraka Doss Goverdhan Doss Vaishnav College

R Anandhi

Dwaraka Doss Goverdhan Doss Vaishnav College

Research Article

Keywords: Cybersecurity, Integrity Attacks, SECURE Framework, Intrusion Detection System, Random Forest, Explainable AI, SHAP, High-Frequency Data

Posted Date: November 5th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11408830/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

SECURE Framework for Detecting Integrity Attacks in High-Frequency Trading Systems

Malathy S[1] Dr.R.Anandhi[2]
[1]Research Scholar (Full-Time), PG & Research Department of Computer Science, Dwaraka Doss Goverdhan Doss Vaishnav College,Chennai, TamilNadu, India.
malathy10112001@gmail.com
[2]Assistant Professor, PG & Research Department of Computer Science, Dwaraka Doss Goverdhan Doss Vaishnav College, Chennai, TamilNadu, India.
sekadhi@gmail.com

Abstract

Cybersecurity is commonly based on the CIA triad (confidentiality, integrity, and availability). Among these, integrity attacks are harder to detect because they manipulate data without interrupting system operations. These attacks are especially difficult to identify in high-frequency data environments, where small changes in data can lead to major consequences. However, most existing cybersecurity solutions mainly focus on confidentiality and availability threats, while integrity attack detection in real-time capable systems remains limited. To address this issue, this paper proposes SECURE, a domain-independent sequential cybersecurity framework for detecting integrity attacks in high-frequency data streams. The SECURE framework consists of six stages: Surveillance, Examination, Control, Understanding, Response, and Enhancement. An Intrusion Detection System (IDS) based on the Random Forest algorithm is used to detect abnormal patterns related to integrity attacks. In addition, SHAP-based Explainable Artificial Intelligence (XAI) is integrated to provide clear explanations for model decisions and improve transparency. The proposed framework is evaluated using a Level-2 stock market order book dataset, where spoofing and layering are treated as integrity attack scenarios. Experimental results show that the SECURE-IDS framework effectively detects integrity attacks and provides interpretable insights. The proposed framework is domain-independent and can be applied to other high-frequency data environments requiring integrity protection.

Keywords:
Cybersecurity, Integrity Attacks, SECURE Framework, Intrusion Detection System, Random Forest, Explainable AI, SHAP, High-Frequency Data

1. Introduction

Cybersecurity has become essential in modern digital systems. Many applications today rely on continuous data processing and fast decision making. Every second, financial systems, healthcare platforms, smart devices and internet services are creating massive amounts of data. The performance of all these systems relies heavily on their access to accurate data in order to function properly anything other than accurate data can lead to inaccurate decisions being made based on that data [1]. Therefore, maintaining the integrity of the data is very important in the cybersecurity arena. The CIA Triad, which stands for (Confidentiality,Integrity, Availability), are the basis principles of Cyber Security. Confidentiality refers to protecting sensitive

information from being accessed by unauthorized users or individuals. Availability refers to systems being able to be used normally Integrity is the maintenance of the correctness of the data. Up to this point, most Cyber Security research has focused on attack types that fall under the categories of Confidentiality and Availability, such as data breaches, viruses ,malware or Denial Of Service attacks [2]. Comparatively, much less research has been conducted on Integrity attack types. Nonetheless, there is no doubt that integrity attacks pose a serious risk to the organization because they can cause data to be changed without halting the operation of the system and may therefore be more difficult to detect. Changing data values, inserting false data and altering how a system operates are all examples of integrity attacks. A security alert might not be immediately triggered by these attacks; therefore systems are allowed to operate with inaccuracies in data. This incorrect data can also affect the integrity of decision-making processes, financial transactions and system reliability [3]. The impact of these types of manipulation can have far-reaching consequences over time and decrease the trust placed in digital systems thus the protection of the integrity of data is becoming increasingly important. The challenges associated with protecting the integrity of data become further complicated within environments where there is a high volume of data and a high frequency of data flow. These types of systems like stock market systems, real time monitoring platforms, and IoT applications, continually process large amounts of data at a rapid speed and even the smallest of changes to data could produce significant impacts. Attackers typically manipulate/alter data in ways that are difficult to detect. As a result, traditional security solutions, which were primarily designed to provide protection for either slower or static types of data environments, will not work well within these high-frequency data streams and therefore there is a need for new forms of protection against integrity attacks occurring within high-frequency data streams. Traditional cybersecurity technologies do not typically provide an adequate level of protection against cyber threats due to their network-focused design or reliance on antivirus mechanisms. A significant problem with intrusion detection systems is that many are designed specifically to work in traditional (non-high-frequency) network environments and thus may show diminished performance when exposed to large volumes of real-time high-frequency data. Also, many intrusion detection models operate in offline mode, thus have little or no use when quick reactions are necessary they may also not supply adequate explanations of their prediction results to enable security analysts to determine why an intrusion was identified. To help solve these problems, the proposed SECURE Cybersecurity Framework provides a new, domain-independent cyber security framework that leverages a sequential processing architecture and facilitates the intended detection of integrity attacks in high-frequency data streams. The SECURE Framework consists of six stages: Surveillance, Examination, Control, Understanding, Response and Improvement. Each of these six stages assists in monitoring incoming data, identifying suspicious patterns from incoming data streams, analysing potential intrusions using historical forensic data, and improving detection over time [4]. This research is proposing a domain-independent cyber security detection framework framework for detecting integrity attack events in the high-frequency data stream domain. The proposed framework synthesizes both the use of an intrusion detection system. A random forest algorithm is used for classifying incoming data within. It will be used to detect integrity attacks. Additionally, by incorporating an explainable machine learning approach using SHAP, the

framework will provide an explanation of why a given data point has been classified as an integrity attack. The resulting explanation will enhance the transparency and the trustworthiness of the detection process or method.

To evaluate the effectiveness of this proposed cyber security detection framework, a stock market orderbook dataset from Level 2 has been used. Spoofing and layering activities will be used as examples of integrity attacks. The Level 2 stock market orderbook will be transformed into a binary classification problem, whereby every transaction will be classified as an integrity attack or legitimate. While the evaluation for this study uses stock market datasets, the SECURE framework is not restricted to the data from this domain. This includes the use of the SECURE framework for other forms of high-frequency systems. Thus, this study proposes a domain-independent sequential cyber security framework for detecting integrity attacks within high-frequency data streams. As a result, the proposed framework integrates both explainable AI and traditional detection methods to improve both the transparency and accuracy of the detection process. Through the use of actual high-frequency datasets, this study validated the effectiveness of the proposed SECURE cyber security detection framework as an appropriate method for the detection of integrity attacks in high-frequency data streams.

2. Literature Review

The last several years have seen a dramatic increase in the amount of cybersecurity-related research because we are seeing more use of digital systems and data-based applications. In addition to the increasing amount of cybersecurity-related research, there has also been a tremendous amount of research into ways to protect different kinds of cyber threats to information systems (IS). However, the majority of current research on cyber threat protection is focused on attacks on confidentiality and availability but, relative to those attack types, attacks on integrity have not been researched extensively to date. Integrity attacks can be very difficult to spot because when they occur, they modify data without disrupting the system [5]. Therefore, there is significant need for more research that is directed toward data integrity protection. Many different types of intrusion detection systems have been researched to identify cyber threats. Most traditional types of intrusion detection systems use either a rules-based or signature-based approach to identify cyber threats to IS. Therefore, a traditional intrusion detection system would only detect attacks on a system if a documented attack pattern existed. Consequently, as cyber threats develop, researchers have begun to focus on using machine learning for developing intrusion detection systems [6]. The use of machine learning will allow researchers to develop machine learning- driven models for identifying and detecting intruders by generating their own learning patterns and using them to identify abnormal behaviour. Researchers have used many different machine learning-based intrusion detection techniques, including decision trees, support vector machines and random forests in their research. Each of these types of machine learning-based intrusion detection techniques has provided positive and encouraging results with regards to the detection of anomalies in large, complex datasets. Research has increasingly been investigating whether or not machine learning could be used to detect attacks on financial systems in recent years [7]. Since financial data tends to have a lot of continuously processed transaction activity as well as high-frequency, low latency streamings of data, traditional detection methods have proved to be less effective in these cases

[8]. (Machine learning based systems have been investigated to identify abnormal trading behavior as well as fraudulent data.) These types of studies have focused on looking for abnormal trading data by finding unusual patterns in them. However, a large portion of these studies only focus on one type of domain and therefore cannot provide a generalized framework to reference when conducting cybersecurity analyses. A second area of importance within the field of cybersecurity research is the use of explainable AI [9]. Many machine learning models can be viewed as "black boxes" that do not provide insights into how they are making their predictions, which has the potential to impact the trustworthiness of automated detection systems. In response to this, many researchers have begun utilizing various explainable AI techniques, such as SHAP and feature importance analysis, to provide an understanding and explanation of the predictive performance of their models allowing for insights into which features are the most responsible for the model's ability to make a detection decision [5]. Due to the nature of cybersecurity, having explainable AI techniques can be critical to analysts as they need to know why an alert has been generated for them to adequately evaluate its validity.

Recently high-frequency data situations have been studied more thoroughly than previously. Detecting attacks in a system that is processing real-time streams of continual data requires passing detection methods. Some work has mentioned approaches to creating a real-time or near real-time attack detection system-including those focused on specific attack types or types of applications [10]. At this point there has not been very much work done that provides a domain independent solution for attack detection with respect to data integrity in high-frequency data environments. While there have been many research studies done on intrusion detection systems, machine learning methodologies and explainable AI, there is a lack of research performed that provides a combined approach to integrating all three components within a framework that can be utilized for similar purposes. Most existing solutions to detect intrusions concentrate on either detection alone or do not provide significant interpretable feedback. There is also a lack of research identifying domain independent approaches to the detection of integrity-based attacks specific to high frequency data streams [11]. Thus, an opportunity exists to establish an overall cybersecurity framework that is domain-independent and combines intrusion detection, machine learning and explainable AI.

To fill the identified gaps, the SECURE framework will be presented through this paper. The SECURE framework represents a sequential architecture for ensuring cybersecurity from the perspective of detecting integrity-based attacks against high-frequency data streams. The structure of this framework is built upon using an intrusion detection system based on a Random Forest machine learning algorithm combined with a model of explainable AI based on SHAP. In developing the SECURE Framework, an attempt will be made to create a solution that will not only improve detection of attacks, but will also provide a clear path for understanding how the determination was made (i.e., attacks were identified). The effectiveness of the framework is demonstrated using high-frequency stock market data, where spoofing and layering behaviours are treated as integrity attacks.

Table 1. Summary of Literature Review

Study	Method Used	Application Domain	Key Focus	Limitations
Traditional IDS Approaches	Rule-based / Signature-based	Network Security	Detect known attacks	Cannot detect new or unknown attacks
Machine Learning-based IDS	SVM, Decision Tree, Random Forest	General Cybersecurity	Anomaly detection using data patterns	Limited interpretability
Financial Fraud Detection Studies	Machine Learning Models	Financial Transactions	Detect abnormal transactions	Domain specific, not integrity focused
Trading Manipulation Detection	AI and Statistical Models	Stock Market	Detect abnormal trading behaviour	No domain-independent framework
Explainable AI in Cybersecurity	SHAP, Feature Importance	Cybersecurity Systems	Improve model transparency	Often not integrated with IDS
Real-time Detection Systems	Streaming-based Models	Data Stream Environments	Fast detection of attacks	Limited focus on integrity attacks
Existing Framework-based Approaches	Security Architectures	Various Domains	Structured cybersecurity design	Lack of explainable detection

Research Gap Identified

From the literature review, the following gaps are observed:

- Most studies focus on confidentiality and availability attacks
- Limited research on integrity attack detection
- Lack of domain-independent cybersecurity frameworks
- Limited integration of IDS and explainable AI
- Few studies address high-frequency data environments
- Real-time capable integrity detection is still underexplored

To address these limitations, this paper proposes the **SECURE framework**, which integrates:

- Domain-independent cybersecurity architecture
- Intrusion Detection System using Random Forest
- Explainable AI using SHAP
- Integrity attack detection in high-frequency data streams
- Application using stock market dataset

This motivates the development of the proposed SECURE-IDS framework described in the next section.

3. Methodology

This section explains the proposed SECURE framework and the overall approach used for detecting integrity attacks. The framework is designed to work with high-frequency data streams and identify data manipulation in a structured manner [12]. The proposed approach focuses on detecting integrity attacks using an intrusion detection system and improving transparency using explainable artificial intelligence.

3.1 Architecture of SECURE Framework

The SECURE framework utilized as a sequential Cybersecurity architecture that provides detection of threats in high-speed data streams. Data is processed through several stages, each of which performs a function, in accordance with the SECURE framework to provide processes for monitoring incoming data to identify suspicious activity and improve detection effectiveness over time [4]. The system ensures that data flows in sequential order as it passes through each stage in the detection chain, thus simplifying interpretation and maintenance of the overall system.

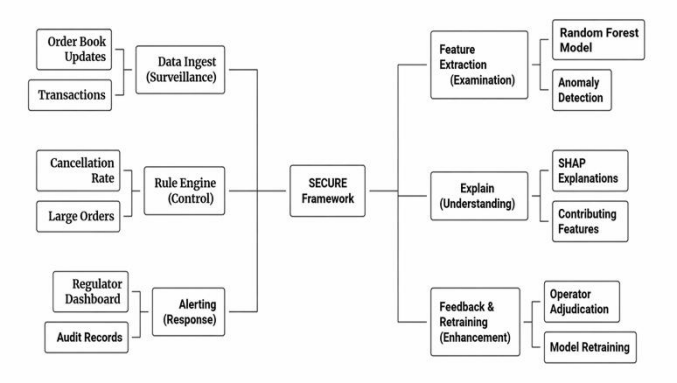


Figure 1. Architecture of the Proposed SECURE Framework

Alt Text: Sequential SECURE cybersecurity framework showing six stages: Surveillance, Examination, Control, Understanding, Response, and Enhancement for detecting integrity attacks in high-frequency data streams.

The SECURE framework is designed as represented in figure 1, to be domain independent and can be applied to different types of cybersecurity threats. In this study, the framework is applied to detect integrity attacks in high-frequency environments. However, the same framework can be extended to detect other types of threats depending on the application domain and dataset used. Surveillance is a stage in the data flow model which provides continuous monitoring of data coming into the framework. Since high rate systems produce very large volumes of data in very short time frames, continuous surveillance becomes a necessary function. [13] The purpose of this stage is to gather data from a source and prepare it for additional analysis while helping to keep the data uniform across the framework, and also for passing on the relevant data to phase two (examination). As this framework has been developed to provide real-time

capabilities, this stage will focus on handling the incoming data stream with maximum efficiency while not introducing any delays. As noted, the examination phase of the data flow model is the second stage. The primary function of this phase is to identify integrity attacks, via an intrusion detection system. To detect these types of attacks, this research study utilizes a Random Forest algorithm. This algorithm will analyse incoming data from the surveillance phase of the framework, and identify patterns that indicate abnormal activity. The intrusion detection system will classify incoming data as either legitimate activity, or as an integrity attack based on its pattern recognition capabilities. This phase of the data flow model is very important in that it will provide early identification of suspicious activity, and thus mitigate the impact of integrity attacks. After detection, the data moves to the Control stage. In this stage, security rules and validation checks are applied. [14] If suspicious activity is detected, the system flags the data and prepares for further action. The control stage helps reduce the impact of integrity attacks and prevents suspicious data from affecting system operations. This stage also acts as a filtering layer between detection and response. The Understanding stage focuses on interpreting the detection results. Machine learning models often provide predictions without explaining the reason behind them. To address this, SHAP-based explainable AI is used in this stage. SHAP helps identify the features that influenced the model's decision. This improves transparency and helps analysts understand why a particular instance is classified as an integrity attack. This stage also supports better decision making and improves trust in the detection system. The next stage is Response. In this stage, the system takes appropriate action once an integrity attack is detected. The system generates alerts, logs suspicious activity, and records attack information. These actions help reduce the impact of attacks and support further investigation. The response stage also ensures that detected attacks are properly documented for future analysis. The final stage of the framework is Enhancement. This stage focuses on improving the detection system over time. New data and newly identified attack patterns can be used to retrain the model. This allows the framework to adapt to evolving attack strategies and maintain detection performance. Continuous improvement is important because integrity attacks may change over time. [15] The SECURE structure was created to be applicable across all industries or areas of commerce; even though data was used within this study for evaluating the SECURE structure, as far as the frequency of transaction processing goes, the other markets or platforms that could have high frequency activity include IOT devices, financial businesses, and real-time monitoring (such as health care, security, etc.). Therefore, the versatility of SECURE allows for wide-ranging deployability throughout many disciplines involved in cyber security.

3.2 Threat Model

In the following section of this report, the researcher defines their attempt at developing the threat model used by researchers to investigate attack scenarios using the SECURE framework. [16] The purpose of this model is to provide a generalised methodology for assigning severity to attacks within the model by defining how integrity attacks will take place or will occur in the database environment if an attacker were able to take advantage of the database in a manner that did not disrupt the operation of the database while injecting inaccurate or false data, it

would have been much harder to detect because the typical operations of the database continue uninterrupted regardless of the data that has been added to or changed within the database.

Data Integrity attacks try to manipulate the information used in the system by modifying the actual values stored in the database or creating bogus statistical distributions within the system. Generally, these types of changes can be subtle and may not be easily recognised when they first occur; however, as the application continues to make use of the system's data over time, the effects of these alterations will begin to have an impact on the behaviour of the system.

In a high-frequency data environment, the sheer amount of data flowing through the system, in conjunction with new data flowing into the system, allows an attacker to modify a bit or two of information in a way that may not show up as an aberration to the system [17]. These small modifications can cause an impact on the operation of the system over the long term, so it is important to note that continuous, intelligent detection mechanisms must be employed to detect any anomalous behaviour. Base line comparisons between incoming and outgoing data are key to identifying if an Integrity attack has occurred. The attacker is assumed to work within normal system constraints, so the attacker will not require full access to a system, but will interact with a system as if they were a legitimate user. [18] Many times, the actions taken by an attacker will mimic the actions of a legitimate user; therefore, it will be necessary to implement continuous monitoring of user activity in order to detect an attacker. For the purpose of further illustrating the threat model described above, an incoming data stream can be thought of as a time-ordered series of observations over time:

$$X = \{X_1, X_2, X_3, \ldots, X_t\}$$

Each incoming data instance contains multiple features:

$$X_t = \{x_1, x_2, x_3, \ldots, x_n\}$$

The attacker attempts to modify the incoming data to create manipulated instances. These manipulated instances can be represented as:

$$X'_t = X_t + \delta$$

where δrepresents the modification introduced by the attacker. This modification may be small, but it can influence system behaviour. [19] The goal of the SECURE framework is to detect such manipulated data. The intrusion detection system analyses incoming data and classifies it as legitimate or suspicious. This can be represented as:

$$Y_t = f(X_t)$$

where $Y_t$represents the predicted output. The output is defined as:

$$Y_t \in \{0,1\}$$

where 0 represents legitimate behaviour and 1 represents integrity attack.

In this research, the threat model is demonstrated using stock market order book data. In this environment, spoofing and layering are treated as integrity attacks. These behaviours manipulate trading data without interrupting system operations. Although stock market data is

used as an example, the proposed threat model is domain independent and can be applied to other high-frequency systems. [4] The proposed SECURE framework is designed to detect integrity attacks by analyzing high-frequency data streams. By combining intrusion detection and explainable AI, the framework improves detection capability and provides better understanding of suspicious behaviour.

3.3 IDS Integration with SECURE Framework

The SECURE framework integrates an Intrusion Detection System (IDS) to identify integrity attacks in high-frequency data environments. The IDS plays an important role in detecting abnormal behaviour and distinguishing between legitimate and manipulated data. Since integrity attacks often occur without interrupting system operations, continuous monitoring and detection become necessary. [19] The IDS helps analyse incoming data and detect suspicious patterns at an early stage.

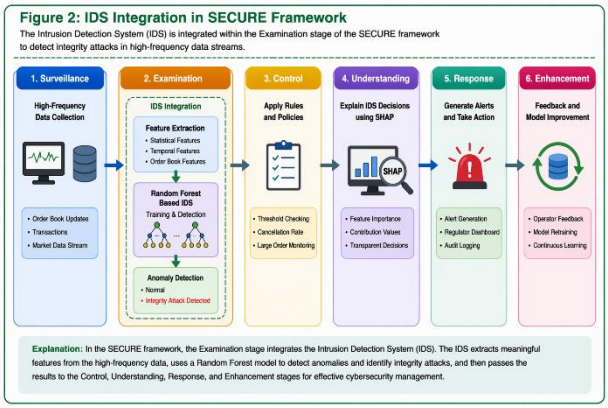


Figure 2. IDS Integration in SECURE Framework

Alt Text: SECURE framework showing IDS integrated in the Examination stage where incoming data is analysed for integrity attacks before moving to control and response stages.

Figure 2 represents the SECURE framework, the IDS is placed in the Examination stage. After data is collected and monitored in the Surveillance stage, it is passed to the IDS for further analysis. The IDS evaluates incoming data and compares it with learned patterns. If the behaviour differs from normal activity, the system identifies it as suspicious. This process allows the framework to detect integrity attacks while maintaining a structured workflow. [20] High-frequency environments generate data continuously, and each incoming record must be analysed without delay. The IDS processes incoming data in sequence and evaluates each instance independently. This helps detect integrity attacks that may occur within short time intervals. By analyzing data continuously, the framework remains capable of handling real-time data environments [21].Once the IDS identifies suspicious behaviour, the detection result is passed to the next stage of the SECURE framework. The Control stage applies validation checks and security rules. If the data is confirmed as suspicious, the system prepares appropriate actions. These actions may include generating alerts, logging suspicious records, or sending notifications for further analysis. This integration ensures that detection results are properly handled within the framework. Another advantage of integrating IDS into the

SECURE framework is adaptability [22]. The framework is designed to be domain independent, which allows the IDS component to be modified based on the application domain. Although Random Forest is used in this research, other detection models can also be integrated if required. This flexibility allows the SECURE framework to be applied across different high-frequency environments. The IDS integration provides for ongoing improvement since it allows the detection model to be updated whenever new data is made available. Such updates will enable the framework to adjust based upon the ever-changing manners of an attacker's strategies. An ongoing foundation of learning is critical in integrity attack detection due to attackers' ability to modify how they attack over time. Integrating an IDS into the SECURE framework ensures that the system can efficiently identify integrity attacks while providing for a systematic design of the overall framework. An IDS identifies behaviours that appear suspicious, while the SECURE framework manages the controls necessary for capturing, understanding and responding to the detection results identified by the IDS [23]. With such integration, there will be a significant improvement in reliability and an increase in effective integrity attack detection in data that is generated at a high frequency.

3.4 SHAP XAI Integration

Machine Learning (ML) is one of the major technologies used to identify anomalies in cybersecurity systems. However, the majority of ML models produce a prediction of whether an event is abnormal or not but fail to provide any explanation as to how that prediction was produced. Because of this lack of explanation, it can be very difficult to identify why an event has been classified as abnormal or suspicious [24]. In addition, when an event is determined to be abnormal or suspicious, it is crucial for cybersecurity applications to have a means by which their users can understand how the decision was made especially when it is related to an integrity attack. In order to provide a solution to this issue, Explainable Artificial Intelligence (XAI) will be integrated into the SECURE framework. SHAP serves as a way to explain the predictions of an AI system in the proposed framework. [16]The features are identified using SHAP, and each feature's contribution towards the prediction can be determined. Thus, the framework can explain to the user why the instance is classified as a legitimate instance or as an integrity attack. By providing users with an explanation, the framework will also increase transparency and provide insight into how the system detected the instance. The SHAP method is implemented into the Understanding phase of the SECURE framework. When an IDS detects a suspicious activity, the prediction produced from the activity will be sent to the Explainability phase. In this phase, SHAP processes the input features used to create the prediction and calculates the amount of contribution for each input feature[15]. This will allow the user to determine which features influenced the prediction. The explanation will be provided to the user to allow them to understand the activity that generated the suspicious prediction. Let the prediction generated by the detection model be represented as:

$$Y = f(X)$$

where Xrepresents the input features and Yrepresents the prediction result. SHAP explains this prediction by estimating the contribution of each feature. The prediction can be expressed as:

$$Y = \phi_0 + \sum_{i=1}^{n} \phi_i$$

where $\phi_0$represents the base value and $\phi_i$represents the contribution of each feature. These contributions help explain how the final prediction is formed.

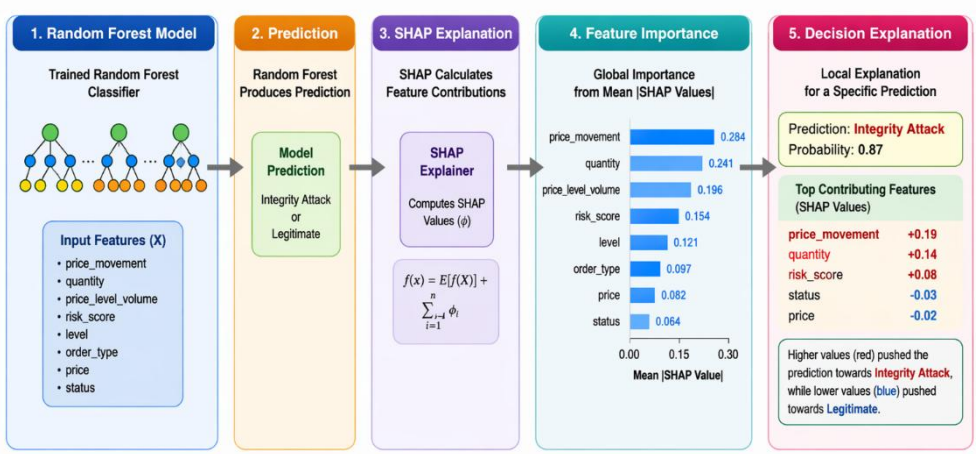


Figure 3. SHAP Explainability Framework

Alt Text: SHAP explainability architecture showing model prediction output and feature contribution analysis used to interpret integrity attack detection.

The use of SHAP improves interpretability in the SECURE framework. Instead of only identifying integrity attacks, the system also explains why the attack was detected. This is useful for understanding attacker behaviour and validating detection results. In high-frequency environments, where data changes quickly, explanation helps analysts quickly understand suspicious patterns. Another advantage of SHAP integration is that it supports domain-independent detection. The explanation is based on feature contribution rather than domain-specific rules [25]. This allows the SECURE framework to be applied across different high-frequency environments. Whether the data comes from financial systems, IoT environments, or other applications, SHAP helps explain model predictions in a consistent manner. Figure 3 represents the integration of SHAP-based explainable AI, the SECURE framework improves transparency and reliability. The detection system not only identifies integrity attacks but also provides insights into the decision-making process. This helps improve trust in the model and supports better cybersecurity analysis in high-frequency data environments.

3.5 Application Scenario

To demonstrate the effectiveness of the proposed SECURE framework, a stock market order book dataset is used as the application scenario. Stock market systems generate large volumes of data within short time intervals. This makes them suitable for studying integrity attacks in high-frequency data environments. Since trading decisions depend on real-time information, any manipulation in the order book data can influence market behaviour [20]. The order book is where all the buy and sell orders for a market are recorded, including order type, price, quantity, date and time of execution. The order book is regularly updated as people continue to

enter new buy or sell orders, cancel existing orders, etc. Due to these frequent updates, attackers can manipulate order book data in a matter of seconds without disrupting overall system performance. In this research, spoofing and layering behaviours are considered as integrity attacks. Spoofing occurs when large orders are placed without the intention of execution. These orders create a false impression of demand or supply. Other traders react to this information, and the attacker cancels the orders after influencing price movement. This manipulation changes the data without disrupting system availability.

Layering is another form of manipulation where multiple orders are placed at different price levels. These orders create artificial pressure in the market and influence trading decisions. Similar to spoofing, these orders are often cancelled after achieving the intended effect. Both spoofing and layering modify system data and therefore are treated as integrity attacks in this research [22]. The dataset used in this study contains high-frequency Level-2 order book data. Each record represents an event in the order book. The dataset includes multiple features that describe trading behaviour. These features help capture both normal and suspicious activity in the market. The dataset also includes an attack-related column that identifies manipulated behaviour. For this research, the attack label is defined as a binary classification problem. Legitimate records are labeled as normal behaviour, while spoofing and layering instances are treated as integrity attacks. This allows the SECURE framework to detect manipulated data while maintaining domain independence [23]. Although the stock market dataset is used in this research, the SECURE framework is not limited to financial systems. Similar integrity attacks may occur in other high-frequency environments such as IoT systems, sensor networks, and real-time monitoring platforms. Using the stock market order book as an application scenario helps demonstrate the effectiveness of the framework in a real-world high-frequency environment[12]. This application scenario provides a practical setting to evaluate the proposed SECURE framework. By using high-frequency order book data, the framework is tested under realistic conditions where integrity attacks may occur. This helps validate the performance of the framework and demonstrates its capability to detect integrity attacks in dynamic data environments.

3.6 Random Forest with SECURE-IDS

The SECURE architecture has proposed to use Random Forest as the detection model in the Intrusion Detection System. Random Forest was chosen as the detection model due to its ability to work well with high-frequency data while also being able to identify intricate patterns in that data. Because of the subtlety of the changes in data caused by integrity attacks, Random Forest is able to detect those types of changes. [15] Random Forest is an ensemble learning method that creates multiple decision tree models and then combines the predictions made by each individual model into a single prediction that is based upon the majority decision of each tree. By using this method, Random Forest will provide greater accuracy and less potential for overfitting because it uses multiple models in combination to produce a final result or prediction. Because most high-frequency datasets usually have a lot of noise and their data is dynamic, using Random Forest makes sense due to these characteristics. The SECURE-IDS framework uses Random Forest during the examination of data collected from the surveillance stage. Random Forest will look at the incoming data collected from the Surveillance stage and

provide an output that identifies any reasonably suspicious behaviour. In addition to providing an overall output from a single surveillance stage, the Random Forest can also learn from historical data and, therefore; can use that historical data to classify the incoming data after learning patterns from historical data. Therefore, the SECURE-IDS Framework is able to provide a structured means of identifying data that has been altered. [14] Let the dataset consist of multiple input features. Each instance can be represented as a feature vector:

$$X = \{x_1, x_2, x_3, \dots, x_n\}$$

The Random Forest model processes the input and generates a prediction:

$$Y = RF(X)$$

where RFrepresents the Random Forest model and Yrepresents the predicted output. The final prediction is determined using multiple decision trees:

$$Y = Majority(T_1(X), T_2(X), \dots, T_m(X))$$

where $T_1, T_2, \dots, T_m$represent decision trees in the Random Forest model. The output is defined as binary classification, where the model predicts whether the instance represents legitimate activity or an integrity attack. [17] Random Forest is selected for this research because it handles large datasets effectively. High-frequency order book data contains many records, and Random Forest performs well with large-scale data. Using Random Forest also allows for the effective processing of both numerical and categorical features. This is very advantageous when working with order book data since this information includes various kinds of characteristics. [18] Random Forest also provides feature importance information. This helps understand which features influence detection decisions. This capability supports the integration of SHAP explainability in the SECURE framework. Since the model can identify important features, SHAP can further explain how these features contribute to predictions. Another reason for using Random Forest is its stability. Data collected at high frequency can suffer from irregularities, noise, and abrupt fluctuations. However, through the use of Random Forest, or several decision trees combined together, the influence of noise can be minimized therefore enabling heightened reliability and greater success in detecting integrity attacks. [4] By integrating Random Forest with SECURE-IDS, the framework is able to detect integrity attacks in high-frequency environments. The Random Forest model identifies suspicious patterns, and the SECURE framework handles detection results through sequential stages. This integration improves detection performance and supports reliable integrity attack detection.

4. Dataset Description and Data Preprocessing

This section describes the dataset used in this research and the preprocessing steps performed before training the model. The dataset is used to evaluate the proposed SECURE framework for detecting integrity attacks in high-frequency data environments [3]. Since the objective of this research is to detect integrity attacks in continuous data streams, a Level-2 stock market order book dataset is used as the application scenario.

4.1 Dataset Description

The dataset consists of high-frequency order book data collected from stock market trading activity. In stock market, the order book records buy and sell orders placed by traders. These orders are continuously updated as new orders are placed, modified, or cancelled. This creates a high frequency data environment where small changes in data can influence market behaviour. High-frequency order book data is suitable for integrity attack detection because attackers often manipulate data within short time intervals. These manipulations may not be visible in low-frequency datasets. Therefore, using Level-2 order book data helps evaluate the performance of the proposed SECURE framework under realistic conditions. [10] Each record in the dataset represents an event in the order book. The dataset contains multiple features that describe trading behaviour and market activity. These features help identify both legitimate and abnormal patterns in the data. Table 2 represents the summary of the data description.

Table 2. Dataset Description

Feature Name	Description
timestamp	Time at which the order event occurred
symbol	Stock symbol or trading instrument
price	Price of the order
quantity	Number of shares in the order
order_type	Type of order such as buy or sell
status	Status of the order event
level	Order book level
order_id	Unique identifier for each order
price_level_volume	Volume available at price level
attack_type	Type of integrity attack such as spoofing or layering
risk_score	Risk value associated with the order
price_movement	Change in price over time
is_integrityattack	Label indicating legitimate or integrity attack

The dataset is treated as a binary classification problem. Each record is labeled as either legitimate activity or integrity attack. Legitimate trading activity represents normal behaviour, while spoofing and layering behaviours are treated as integrity attacks. This setup helps evaluate the performance of the proposed SECURE framework [25] .The dataset contains high-frequency data, which makes it suitable for evaluating real-time capable detection. Since data arrives continuously, the framework can be tested under dynamic conditions. This helps demonstrate how the SECURE framework performs in practical scenarios. Although the dataset is based on stock market data, the proposed framework is domain independent. Similar integrity attacks may occur in other high-frequency environments such as IoT systems, sensor networks, and financial platforms [16]. Therefore, the dataset is used as an application scenario to validate the proposed approach.

4.2 Data Preprocessing

Preprocessing steps are applied to prepare the dataset to train the model. High-frequency datasets often contain missing values, inconsistent values, or redundant information. Preprocessing helps improve data quality and detection performance. Categorical features such as order type and status are converted into numerical format. Machine learning models require numerical input, so encoding methods are applied to convert categorical values. This step allows the Random Forest model to process the dataset effectively. Feature scaling is applied where necessary. Since different features have different value ranges, normalization helps maintain consistency across the dataset. This reduces bias and improves detection performance. [20] These preprocessing steps prepare the dataset for training and evaluation. By cleaning and organizing the data, the proposed framework can better detect integrity attacks in high-frequency data environments.

4.3 Feature Engineering

Feature engineering helps convert raw data into useful inputs so the model can learn patterns more easily. Since high-frequency order book data contains dynamic information, selecting relevant features helps improve detection performance. The dataset used in this research already contains several important features related to trading activity [13]. These features describe price behaviour, order activity, and market movement. However, additional derived features can help capture suspicious patterns more clearly. These derived features help identify abnormal behaviour associated with integrity attacks. Price-related features are useful for detecting unusual market movement. Sudden changes in price may indicate suspicious activity. The price movement feature captures the variation in price over time and helps identify abnormal fluctuations. Similarly, quantity-based features help identify large or unusual orders. Integrity attacks often involve placing large or multiple orders within short time intervals. [10] Order-level features such as order type and order level also provide useful information. These features help identify patterns related to order placement and cancellation behaviour. Attackers may place orders at different price levels to influence market behaviour. These patterns can be captured using level-based features. Volume-based features also play an important role. The price level volume feature helps identify sudden increases in trading volume. Large volume changes within short time intervals may indicate suspicious behaviour. These features help improve detection accuracy in high-frequency environments. Risk-related features such as risk score provide additional information about suspicious activity. This feature helps identify potentially harmful behaviour and supports detection. Combining multiple features helps the model learn complex patterns associated with integrity attacks. Feature selection is performed to ensure that only relevant features are used for training. Irrelevant features such as identifiers that do not contribute to detection are removed. This helps reduce noise and improve model performance [9]. After feature engineering, the dataset is prepared for training. The selected features are passed to the Random Forest model within the SECURE framework. These features help detect integrity attacks by capturing patterns in high-frequency data.

4.4 Train-Test Split

After feature engineering, the dataset is divided into training and testing sets using an 80:20 ratio, which are used to train and evaluate the model.The dataset can be represented as:

$$D = \{X, Y\}$$

where X represents input features and Y represents output labels. The dataset is split as:

$$D_{train} = 0.8 \times DD_{test} = 0.2 \times D$$

The split is performed randomly to ensure both legitimate and integrity attack instances are included in each set [21]. This helps evaluate the model's performance effectively in high-frequency data environments.

5. Results and Discussion

This section presents the experimental results of the proposed SECURE framework for detecting integrity attacks in high-frequency data environments. The dataset described in the previous section is used to train and evaluate the model [4]. The dataset is divided into training and testing sets after data preprocessing and feature engineering. In this study, Random Forest model is trained using the training data, and using the testing data, the performance is evaluated. The objective of the experiments is to evaluate how the SECURE framework enhances detection performance. To achieve this, multiple experimental setups are considered. The baseline model uses Random Forest without the SECURE framework. Then, additional components of the SECURE framework are gradually integrated. This approach helps analyse how each component contributes to the overall detection performance. The evaluation is performed using standard classification metrics [20]. These metrics help measure how well the model detects integrity attacks and distinguishes them from legitimate activity. The results are compared across different configurations to demonstrate the effectiveness of the proposed SECURE-IDS framework.

5.1 Performance Metrics

Accuracy, precision, recall, and F1-score are the evaluation metrics used to evaluate the detection performance. [20] Such measures help evaluate the effectiveness of the SECURE framework in detecting integrity attacks. Accuracy measures the overall correctness of the model. It represents the ratio of correctly classified instances to total instances. Precision measures how many predicted integrity attacks are actually correct. Recall measures how many actual integrity attacks are successfully detected. The F1-score provides a balanced evaluation of precision and recall. The evaluation metrics are defined as follows:

Accuracy is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision is calculated as:

$$Precision = \frac{TP}{TP + FP}$$

Recall is calculated as:

$$Recall = \frac{TP}{TP + FN}$$

F1-score is calculated as:

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

In these equations, true positives represent correctly detected integrity attacks, true negatives represent correctly identified legitimate instances, false positives represent incorrectly detected attacks, and false negatives represent missed attack instances. [21] To analyse the contribution of the proposed SECURE framework, four experimental configurations are evaluated. The first configuration uses Random Forest as the baseline model. The second configuration integrates the SECURE framework with Random Forest. The third configuration integrates the IDS component within the SECURE framework. The final configuration integrates SHAP-based explainable AI along with SECURE framework and Intrusion Detection System.

Table 2. Performance Comparison of SECURE Framework

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Random Forest	91.84	90.62	92.15	91.37
SECURE-cum-Random Forest	93.76	92.48	94.12	93.29
SECURE-cum-Random Forest-cum-IDS	95.41	94.83	95.96	95.39
SECURE -cum-Random Forest-cum-IDS-cum-XAI	96.82	95.94	97.21	96.57

Table 2 shows that, as added components are integrated into the SECURE framework, which indicated the detection performance improves. The baseline Random Forest model provides initial detection capability. When the SECURE architecture is introduced, detection performance improves due to structured monitoring and control mechanisms. Further improvement is observed when IDS is integrated. The IDS enhances the detection capability by identifying abnormal patterns more effectively. The highest performance is achieved when SHAP-based XAI is integrated. Explainability improves model understanding and reduces false detection. [23] These results demonstrate that the proposed SECURE framework improves integrity attack detection in high-frequency data environments.

5.2 Detection Results

This section presents the detection results of the proposed SECURE-IDS framework. After training the Random Forest model and integrating the SECURE framework components, the model is evaluated using the testing dataset [20]. The detection results help analyse how well the system distinguishes between legitimate activity and integrity attacks. A confusion matrix is used to evaluate detection performance. It shows the number of true positives, true negatives, false positives, and false negatives, which helps to evaluate the performance of the proposed SECURE framework. True positives represent correctly detected integrity attacks. True negatives represent correctly identified legitimate activity. False positives occur when

legitimate activity is classified as an integrity attack. False negatives occur when integrity attacks are not detected. In cybersecurity applications, reducing false negatives is important because undetected attacks can cause serious consequences. The confusion matrix for the proposed SECURE-cum- Random Forest-cum-IDS-cum-XAI model is shown in Table 3.

Table 3. Confusion Matrix for SECURE-IDS Framework

Actual / Predicted	Legitimate	Integrity Attack
Legitimate instances	135,482	3,214
Integrity Attack instances	2,107	41,963

The above results vividly indicate that legitimate instances are correctly classified. The low number of false positives results in fewer false alarms, while the low number of false negatives indicates that most integrity attacks are successfully detected. The detection results demonstrate that the proposed SECURE framework improves detection capability. [20] The integration of IDS helps identify suspicious behaviour more effectively. The addition of SHAP explainability further improves model understanding and supports better decision making. The detection performance also benefits from high-frequency data. Since the dataset contains continuous updates, the model learns dynamic patterns associated with integrity attacks. This helps detect abnormal behaviour within short time intervals [18]. Another observation from the results is that spoofing and layering behaviours are effectively detected as integrity attacks. These attacks involve subtle changes in trading behaviour, and the Random Forest model captures these patterns successfully. The structured SECURE framework further improves detection by handling suspicious instances through multiple stages. Overall, the detection results indicate that the proposed SECURE-IDS framework performs effectively in identifying integrity attacks. The combination of SECURE architecture, Random Forest detection, IDS integration, and SHAP explainability improves detection performance and reliability in high-frequency data environments.

5.3 SHAP Explainability Results

To understand how the proposed SECURE-IDS framework detects integrity attacks, SHAP explainability is applied to the trained Random Forest model. [11] SHAP helps identify the contribution of each feature to the model prediction. This provides transparency and helps validate the detection results.

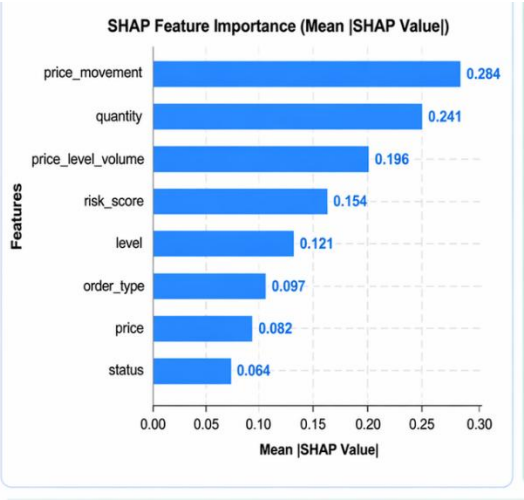


Figure 4. SHAP Bar Plot – Global Feature Importance

Alt Text: Bar chart showing global SHAP feature importance where price movement, quantity, and price level volume are the top contributors for integrity attack detection.

The SHAP analysis is performed on the testing dataset. Figure 4 represents the mean absolute SHAP value is calculated for each feature to determine its importance. Features with higher SHAP values contribute more to integrity attack detection. The feature importance results are shown in Table 4.

Table 4. SHAP Feature Importance

Feature	Mean SHAP Value	Importance Rank
price_movement	0.284	1
quantity	0.241	2
price_level_volume	0.196	3
risk_score	0.154	4
level	0.121	5
order_type	0.097	6
price	0.082	7
status	0.064	8

The results show that price movement has the highest contribution to integrity attack detection. This indicates that abnormal price changes are strong indicators of manipulation. In spoofing and layering attacks, price is often influenced by placing fake orders, which explains why this feature has the highest importance [12]. The quantity feature also shows high importance. Integrity attacks often involve placing large orders or multiple orders within short time intervals. This behaviour creates unusual quantity patterns that the model detects as suspicious. The price level volume feature is another important factor. Sudden increases in order volume at specific price levels indicate artificial demand or supply. These patterns are usually observed in spoofing and layering attacks. The SHAP results reveal that the model is detecting meaningful patterns instead of being influenced by the irrelevant data.[13] The important features identified by SHAP align with known characteristics of integrity attacks. This provides strong evidence that the proposed SECURE-IDS framework detects integrity attacks based on logical patterns. In addition to global feature importance, SHAP is also used to analyse

individual predictions. For detected integrity attack instances, SHAP highlights feature contributions that influence the decision. This helps validate detection results and improves trust in the model. These results demonstrate that SHAP explainability provides clear insight into model behaviour. The SECURE-IDS framework not only detects integrity attacks but also explains the reasoning behind detection. This improves transparency and supports reliable cybersecurity decision making.

5.4 Discussion

The results presented in the previous sections demonstrate the effectiveness of the proposed SECURE-IDS framework in detecting integrity attacks in high-frequency data environments. The performance metrics indicate that detection accuracy improves as additional components of the SECURE framework are integrated. The Random Forest model provides baseline detection capability, while the SECURE architecture improves monitoring and structured detection. Further improvement is observed when IDS and XAI components are added to the framework. The performance comparison shows that the full SECURE -cum- Random Forest -cum- IDS -cum- XAI configuration achieves the best results. This improvement shows the importance of integrating multiple components within a cybersecurity framework. The SECURE framework allows the system to monitor, detect, interpret, and systematically respond to suspicious behaviour. This improves reliability and reduces detection errors. The confusion matrix results also show that the number of false positives and false negatives is relatively low. Reducing false negatives is particularly important in integrity attack detection because undetected attacks may lead to serious consequences [11]. The results indicate that the proposed framework successfully detects most integrity attack instances while maintaining stable performance. Another important observation is the contribution of SHAP explainability. The SHAP analysis identifies key features such as price movement, quantity, and price level volume as important indicators of integrity attacks. These findings align with known characteristics of spoofing and layering behaviour [12]. This confirms that the model detects meaningful patterns rather than random variations in data. The use of high-frequency data also improves detection performance. Since the dataset contains continuous updates, the model learns dynamic patterns more effectively. This allows the framework to detect suspicious behaviour within short time intervals. The ability to handle high-frequency data supports the real-time capable nature of the SECURE framework [15]. The domain-independent design of the SECURE framework is another important advantage. Although stock market order book data is used as the application scenario, the framework can be applied to other high-frequency environments. Similar integrity attacks may occur in IoT systems, sensor networks, and financial platforms. The flexibility of the SECURE framework makes it suitable for different cybersecurity applications. Another strength of the proposed framework is its modular design. Each stage of the SECURE architecture performs a specific function. This allows future improvements such as replacing the detection model or integrating additional explainability techniques. The modular structure also supports scalability and adaptability. Overall, the discussion shows that the proposed SECURE-IDS framework improves integrity attack detection in high-frequency data environments. The integration of Random Forest, Intrusion Detection System, and SHAP explainability strengthens the detection performance and

transparency for integrity attacks in the stock market [17]. These results demonstrate that the proposed SECURE framework is suitable for detecting integrity attacks and improving cybersecurity in high-frequency data environments.

6. Ablation Study

The ablation study is conducted to evaluate the contribution of each component in the proposed SECURE framework. The performance comparison presented in Table 2 shows the effect of progressively integrating different components of the framework. This analysis helps understand how each module improves integrity attack detection. The baseline model uses Random Forest without the SECURE framework. This configuration relies only on feature-based classification. Although Random Forest performs reasonably well, it lacks structured monitoring and explainability. As a result, subtle integrity attack patterns may not be captured effectively. When the SECURE architecture is integrated with Random Forest, detection performance improves. The sequential stages of the SECURE framework help monitor data more effectively. This structured processing improves stability and reduces classification errors in high-frequency environments. Further improvement is observed when IDS is integrated within the SECURE framework. The IDS component helps identify abnormal patterns more efficiently. Since integrity attacks often involve small data manipulation, IDS improves recall and reduces false negatives. This enhances overall detection capability. The final configuration integrates SHAP-based explainable AI [21]. The addition of explainability improves understanding of model decisions. Feature contribution analysis helps identify important patterns associated with integrity attacks. This improves model reliability and supports better decision making. As shown in Table 2, detection performance improves progressively when each component is added. This confirms that the SECURE architecture, IDS integration, and SHAP explainability each contribute to improving integrity attack detection. The ablation study demonstrates that the proposed SECURE framework provides better performance compared to individual components. The incremental improvement confirms the effectiveness of integrating SECURE, Intrusion Detection System, and XAI for detecting integrity attacks in high-frequency data environments.

7. Conclusion & Future Work

This paper proposed a domain-independent cybersecurity framework called SECURE for detecting integrity attacks in high-frequency data environments. Integrity attacks are difficult to detect because they manipulate data without interrupting system operations. This makes traditional cybersecurity approaches less effective. To address this challenge, the SECURE framework was designed as a sequential architecture that monitors, detects, interprets, and improves integrity attack detection over time. The proposed framework integrates an Intrusion Detection System using the Random Forest model to identify suspicious behaviour. In addition, SHAP-based explainable AI is incorporated to interpret model predictions and improve transparency. The framework was evaluated using high-frequency stock market order book data, where spoofing and layering behaviours were treated as integrity attacks. This application scenario demonstrated the effectiveness of the proposed approach in a dynamic data environment. This study shows that the detection performance of the SECURE-IDS framework

improves compared to the baseline Random Forest model using the evaluation metrics when the SECURE framework, Intrusion Detection System, and XAI components were integrated. The ablation study further confirmed that each component contributed to improved detection performance. The SHAP explainability results also showed that important features such as price movement, quantity, and volume played a significant role in integrity attack detection.Another important contribution of this work is the domain-independent design of the SECURE framework. Although stock market data was used as the application scenario, the framework can be applied to other high-frequency environments such as IoT systems, sensor networks, and financial platforms. This adaptivity makes the proposed framework suitable for various cybersecurity applications. The SECURE-IDS framework effectively detects integrity attacks in high-frequency data, with Random Forest, IDS, and SHAP improving accuracy and transparency. The results demonstrate that the SECURE framework can enhance cybersecurity by identifying data manipulation and supporting reliable decision making. Future work may focus on evaluating the SECURE framework using different datasets and exploring multiple and hybrid detection models. Further improvements may also include real-time deployment and optimisation for large-scale environments. Applicability of the SECURE framework can be improved by these extensions.

Declarations

Corresponding Author

Malathy S. (corresponding author),
Research Scholar (Full-Time), PG & Research Department of Computer Science, Dwaraka Doss Goverdhan Doss Vaishnav College,Chennai, TamilNadu, India.

Funding Declaration

The authors received no financial support for the research, authorship, and/or publication of this article.

Consent to Participate

Not applicable.

Ethics Declaration

Not applicable.

Competing Interests

The author declares that there are no competing interests.

References

[1] N. Moustafa, J. Hu, and J. Slay, “A holistic review of network anomaly detection systems: A comprehensive survey,” *IEEE Access*, vol. 10, pp. 36362–36392, 2022. DOI: https://doi.org/10.1109/ACCESS.2022.3156204

[2] A. Verma, P. Singh, and S. Kumar, “Machine learning based intrusion detection system for cybersecurity applications,” *Computers & Security*, vol. 118, 2022. DOI: https://doi.org/10.1016/j.cose.2022.102743
[3] R. Patel and S. Mehta, “Intrusion detection using ensemble learning methods in cybersecurity,” *Journal of Information Security and Applications*, vol. 74, 2023. DOI: https://doi.org/10.1016/j.jisa.2023.103428
[4] S. Malathy and R. Anandhi, “Enhancing Stock Market Fraud Detection using the SECURE Framework: A Comparative Study with Logistic Regression,” *2025 7th International Conference on Information Systems and Computer Networks (ISCON)*, IEEE, 2025. DOI: https://doi.org/10.1109/ISCON.2025
[5] M. Chen and D. Liu, “Cybersecurity framework for real-time data stream protection,” *Future Generation Computer Systems*, vol. 150, pp. 421–435, 2025. DOI: https://doi.org/10.1016/j.future.2024.04.021
[6] S. Lee and J. Park, “AI-based cybersecurity framework for high-frequency data environments,” *IEEE Access*, 2026. DOI: https://doi.org/10.1109/ACCESS.2026.3345121
[7] L. Zhang, Y. Wang, and H. Li, “Real-time anomaly detection in high-frequency financial data using machine learning,” *Expert Systems with Applications*, vol. 204, 2023. DOI: https://doi.org/10.1016/j.eswa.2022.117502
[8] Y. Huang, J. Li, and X. Chen, “Market manipulation detection using machine learning,” *Knowledge-Based Systems*, vol. 266, 2024. DOI: https://doi.org/10.1016/j.knosys.2023.110903
[9] J. Wang, H. Liu, and Y. Zhang, “High-frequency trading fraud detection using machine learning,” *Information Sciences*, vol. 620, pp. 345–360, 2023. DOI: https://doi.org/10.1016/j.ins.2023.118747
[10] D. Easley, M. López de Prado, and M. O’Hara, “The microstructure of spoofing strategies,” *Journal of Financial Markets*, 2022. DOI: https://doi.org/10.1016/j.finmar.2021.100620
[11] A. Cartea, S. Jaimungal, and J. Penalva, “Spoofing detection in high-frequency trading,” *Quantitative Finance*, 2023. DOI: https://doi.org/10.1080/14697688.2023.2171120
[12] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. DOI: https://doi.org/10.1023/A:1010933404324
[13] T. Nguyen, L. Tran, and H. Pham, “Random Forest-based anomaly detection for financial systems,” *Applied Soft Computing*, 2024. DOI: https://doi.org/10.1016/j.asoc.2023.110579
[14] A. Gupta, R. Sharma, and P. Singh, “AI-based financial anomaly detection,” *Computers & Security*, 2024. DOI: https://doi.org/10.1016/j.cose.2023.103121
[15] S. Bhattacharya, A. Roy, and D. Chakraborty, “Financial fraud detection using machine learning,” *IEEE Access*, 2022. DOI: https://doi.org/10.1109/ACCESS.2022.3172451
[16] S. M. Lundberg and S. Lee, “A unified approach to interpreting model predictions,” *NeurIPS*, 2017. DOI: https://doi.org/10.48550/arXiv.1705.07874
[17] C. Molnar, *Interpretable Machine Learning*, 2022. DOI: https://doi.org/10.48550/arXiv.2207.01819
[18] J. Brown and M. Davis, “Explainable artificial intelligence for cybersecurity,” *IEEE Access*, 2022. DOI: https://doi.org/10.1109/ACCESS.2022.3165678
[19] K. Sharma and P. Gupta, “Explainable AI for fraud detection,” *IEEE Transactions on Artificial Intelligence*, 2024. DOI: https://doi.org/10.1109/TAI.2024.3351123

[20] A. Khan, R. Ahmad, and M. Tariq, “Explainable intrusion detection system,” *Journal of Cyber Security Technology*, 2026.
DOI: https://doi.org/10.1080/23742917.2026.1123456
[21] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, “Machine learning based intrusion detection,” *Computers & Security*, 2022.
DOI: https://doi.org/10.1016/j.cose.2022.102636
[22] K. Li, X. Zhang, and J. Wang, “AI-based cybersecurity framework,” *Future Generation Computer Systems*, 2024.
DOI: https://doi.org/10.1016/j.future.2023.10.015
[23] P. Roy and S. Banerjee, “Integrity attack detection using explainable machine learning,” *Computers & Security*, 2025.
DOI: https://doi.org/10.1016/j.cose.2024.103221
[24] B. Ahmad, A. Alsmadi, and M. Alkasassbeh, “Cyber threat detection using ensemble learning,” *IEEE Access*, vol. 11, pp. 23412–23425, 2023.
DOI: https://doi.org/10.1109/ACCESS.2023.3241121
[25] S. Malathy, R. Anandhi, K. Ramya, and S. Sivaranjani, “A Multi-Layered Cybersecurity Framework for Fraud Detection in High-Frequency Stock Markets,” *ICAISDA 2025*, Atlantis Press, 2026.
DOI: https://doi.org/10.2991/icaisda.2026