ANOMAL DETECTION IN ONLINE BETTING

ABSTRACT

Anomaly detection in online betting involves identifying unusual patterns or behaviors that deviate from the norm. It aims to detect fraudulent activities, suspicious betting patterns, or abnormal transactions that could indicate cheating or manipulation. Online sports betting platforms generate massive volumes of real-time data involving teams, leagues, and match outcomes. Such large-scale data environments are often vulnerable to irregular or fraudulent betting activities, making anomaly detection a vital task to ensure fairness, transparency, and integrity in the betting ecosystem.

This project presents a deep learning based anomaly detection framework using football match data to identify unusual betting behaviors and suspicious outcomes. Four advanced models like Autoencoder, Graph Neural Network (GNN), League Embedding Model and Deep Feedforward Neural Network (DFFNN) were implemented and compared based on their performance metrics, including accuracy, precision, recall, and F1-score.

Among all models, the League Embedding Model achieved the best performance with an accuracy of 92%, effectively detecting anomalies across teams and leagues. Further, team-based (country-wise) analyses were conducted to visualize the distribution of anomalies and to identify potential regions or teams with irregular betting patterns.

The results demonstrate that integrating embedding and graph-based deep learning techniques significantly enhances anomaly detection in complex sports data. This framework contributes to developing more secure, reliable, and data-driven online betting monitoring systems.

KEYWORDS: Online betting, Anomaly detection, Autoencoder, Graph Neural Network, League Embedding Model, Deep Feedforward Neural Network.

CHAPTER – I

INTRODUCTION

INTRODUCTION

In the digital era, **online sports betting** has emerged as one of the most dynamic and data-driven sectors within the global entertainment industry. Sports betting platforms continuously generate enormous volumes of transactional data, ranging from odds fluctuations and match outcomes to user betting behaviors. Among all sports, **football (soccer)** stands out as the most popular and frequently wagered sport worldwide, leading to the accumulation of extensive historical and live betting datasets.

While the growth of online betting platforms has created new economic opportunities, it has also introduced serious challenges such as **fraudulent activities**, **match-fixing**, **abnormal betting trends**, and **data manipulation**. These anomalies can distort fair play, mislead betting predictions, and even damage the credibility of sports organizations and betting platforms. Therefore, detecting anomalies in betting data is crucial for ensuring transparency, integrity, and ethical governance in sports betting ecosystems.

This project proposes a **deep learning–based anomaly detection framework** for online football betting systems. The framework leverages multiple neural architectures including **Autoencoder**, **Graph Neural Network (GNN)**, **Deep Feedforward Neural Network (DFFNN)**, and **League Embedding Model** to identify anomalies in football betting data. Each model explores a different perspective of the data:

- **Autoencoder** focuses on reconstructing input data to detect deviations between expected and actual patterns.
- **GNN** captures the graph-based relationships between football teams (e.g., home vs. away matches).
- **League Embedding Model**, the most advanced, combines team and league embeddings to understand both team-level and league-level dependencies for anomaly classification.
- **DFFNN** serves as a robust baseline deep network for learning general betting trends.

The dataset used in this study consists of labelled football match data containing odds, goal differences, league identifiers, and a binary indicator specifying whether a match instance is normal or anomalous. Data preprocessing steps include **missing value handling**, **scaling**, and

encoding categorical identifiers into numerical embeddings to ensure compatibility with neural network inputs.

Each model was evaluated using multiple performance metrics including **accuracy**, **precision**, **recall**, and **F1-score** to provide a comprehensive understanding of their predictive capabilities. The results revealed that the **League Embedding Model** achieved the highest classification performance, with approximately **92% accuracy**, outperforming the other architectures.

Beyond classification, the study also incorporated **Team-Based Analysis (League wise) Analysis** to interpret how anomalies are distributed across teams and regions. These analyses help uncover potentially suspicious patterns, such as teams or countries exhibiting disproportionately high anomaly frequencies, which could indicate irregular betting or performance behaviors.

Overall, this project demonstrates that integrating **deep learning**, **graph structures**, and **embedding techniques** provides a powerful, data-driven approach for detecting anomalies in sports betting. The proposed framework contributes to building **intelligent, transparent, and secure betting systems**, ensuring fairness and accountability within the rapidly expanding online sports betting ecosystem.

1.1 PROJECT OVERVIEW

The rise of **online sports betting platforms** has transformed how people engage with sporting events, especially in **football**, which attracts millions of bettors worldwide. These platforms generate vast amounts of betting and match data, creating challenges in maintaining **fairness and integrity**. Detecting **anomalous betting patterns**, such as irregular odds or suspicious match outcomes, is essential to prevent fraud and ensure transparency in the betting ecosystem.

This project introduces a **deep learning–based anomaly detection system** designed to identify abnormal patterns in football betting data. Four models like **Autoencoder**, **Graph Neural Network (GNN)**, **Deep Feedforward Neural Network (DFFNN)**, and **League Embedding Model** were developed and compared to classify matches as *normal* or *anomalous*. Among these, the **League Embedding Model** achieved the highest performance with **92% accuracy**, proving highly effective in capturing team- and league-level relationships. The project also includes **team-based** analyses to visualize anomaly distributions and highlight potential regions or teams with unusual activity.

Overall, this work demonstrates that integrating graph structures with embedding techniques significantly improves anomaly detection in sports data, supporting fraud prevention, sports integrity monitoring, and data-driven decision-making in online betting systems.

1.2 MOTIVATION

With the rapid growth of online sports betting, particularly in football, there has been a massive increase in the volume of data generated from betting platforms. These include real-time odds, player statistics, match outcomes, and user betting behaviors. While this digital transformation has made betting more engaging and accessible, it has also introduced challenges such as fraudulent activities, match-fixing, and irregular betting patterns that threaten the fairness and integrity of the sports industry. Detecting such anomalies is therefore crucial for maintaining transparency and ensuring the trustworthiness of betting ecosystems.

By analyzing structured football match data, the system can effectively classify whether a match event is normal or anomalous, enabling betting regulators, analysts, and organizations to make informed, proactive decisions. Beyond fraud detection, the model also aids in understanding betting behaviours, identifying high-risk teams or leagues, and improving sports integrity monitoring practices.

Furthermore, this project aligns with the broader vision of applying artificial intelligence to promote ethical and transparent digital ecosystems. As anomaly detection becomes increasingly vital in areas like finance, cybersecurity, and online gaming, developing a scalable and interpretable detection system contributes to responsible AI deployment in the sports betting industry. By combining advanced neural architectures with real-world data analysis, this research aims to enhance both the security and credibility of global online betting platforms.

1.3 DATASET

The **Football Betting Odds Dataset** serves as the foundation for detecting anomalies in online football betting activities. This dataset provides detailed information about football matches, betting odds, and outcome probabilities, which are essential for understanding and identifying irregular betting patterns that may indicate fraudulent behavior or manipulated events.

The dataset is stored in a **CSV file** and includes records from various football leagues and teams. Each record corresponds to a single football match, containing attributes that represent match outcomes, implied betting probabilities, team identities, and derived statistical measures such as goal differences and surprise factors. These features allow deep learning models to learn normal betting behaviours and distinguish them from suspicious or abnormal patterns.

Before analysis, the dataset was **preprocessed** by handling missing values, normalizing numerical features, and encoding categorical variables (such as team and league identifiers) into numerical form to ensure compatibility with neural network models. The data was further divided into **training and testing sets** to evaluate model performance objectively.

Feature	**Description**
home_team	Name of the home team participating in the match.
away_team	Name of the away (opponent) team.
goal_difference	Difference between goals scored by home and away teams.
implied_prob_home	Implied probability of the home team winning based on betting odds.
implied_prob_draw	Implied probability of a draw outcome.
implied_prob_away	Implied probability of the away team winning.
surprise_factor	A derived metric indicating unexpected outcomes or unusual betting patterns.
league	The league or competition where the match took place.
country	Country associated with the league (used for spatial analysis).
is_anomaly	Binary label indicating match status — **0 = Normal**, **1 = Anomalous**.

CHAPTER - II

LITERATURE SURVEY

LITERATURE SURVEY

This chapter provides a comprehensive review of existing studies and methodologies related to anomaly detection in online football betting using deep learning. It examines the evolution of detection techniques and the use of advanced neural models such as Autoencoders, Graph Neural Networks, League Embedding Models, and Deep Feedforward Neural Networks.

2.1 THE PROBLEM OF ANOMALY DETECTION IN ONLINE FOOTBALL BETTING USING DEEP LEARNING

Online football betting has grown into a massive data-driven industry, generating large-scale transactional and behavioral datasets. However, this expansion has also increased the prevalence of fraudulent practices such as odds manipulation, match-fixing, and irregular betting patterns (FIFA, 2023). Traditional statistical methods often struggle to identify such irregularities due to the complex, nonlinear, and dynamic nature of betting data (Chalapathy & Chawla, 2019; Zhang & Chen, 2020).

Anomaly detection in this context focuses on identifying patterns that deviate significantly from normal behavior such as unexpected betting trends or suspicious outcome correlations (Li & Zhang, 2021). Deep learning provides a scalable and data-driven solution capable of automatically learning intricate feature representations and hidden dependencies within football betting data (Goodfellow, Bengio, & Courville, 2016).

By leveraging advanced neural architectures, deep learning enables adaptive learning and continuous anomaly monitoring, offering superior accuracy and robustness compared to traditional detection methods (Paszke et al., 2019; OpenAI, 2025). This makes it a powerful tool for maintaining transparency and fairness in online betting platforms.

2.2 DEEP LEARNING APPROACHES FOR ANOMALY DETECTION IN ONLINE FOOTBALL BETTING

Deep learning has revolutionized anomaly detection by allowing models to automatically extract meaningful features from complex and high-dimensional data (Chalapathy & Chawla, 2019). In the sports betting domain, various neural architectures have been employed to detect fraudulent or suspicious activities by learning patterns in match statistics, betting odds, and implied probabilities (Zhang & Chen, 2020).

The most significant deep learning models used for this purpose include:

- **Autoencoder (AE):** An unsupervised neural model that learns to reconstruct normal data patterns and identifies anomalies through reconstruction error (Hinton & Salakhutdinov, 2006; Kingma & Welling, 2014).
- **Graph Neural Network (GNN):** A relational deep learning model that represents teams and matches as nodes and edges, respectively, to capture inter-team dependencies (Kipf & Welling, 2017; Zhou et al., 2020).
- **League Embedding Model (LEM):** A hybrid embedding-based model that integrates team, league, and match-level features to learn rich contextual relationships (Li & Zhang, 2021).
- **Deep Feedforward Neural Network (DFFNN):** A multilayer neural model that learns nonlinear mappings between input features and anomaly labels (Goodfellow et al., 2016; Srivastava et al., 2014).

These models collectively enhance the detection of subtle, contextual, and data-driven anomalies in complex football betting environments.

2.3 ROLE OF AUTOENCODER

The **Autoencoder** plays a crucial role in unsupervised anomaly detection due to its ability to learn compact representations of normal data distributions (Hinton & Salakhutdinov, 2006). The model consists of two components an **encoder** that compresses input features into latent representations, and a **decoder** that reconstructs the original data.

During training, the Autoencoder minimizes reconstruction loss for normal samples. When anomalous data is input, reconstruction error increases significantly because the model cannot accurately reproduce unseen or irregular patterns (Kingma & Welling, 2014).

In online football betting, Autoencoders effectively learn normal betting behavior such as expected odds variations and common match outcomes while flagging unusual trends as anomalies (Zhang & Chen, 2020; Li & Zhang, 2021). This makes Autoencoders particularly valuable in settings with limited labeled data (Chalapathy & Chawla, 2019).

2.3.1 ROLE OF GRAPH NEURAL NETWORK (GNN)

The **Graph Neural Network (GNN)** extends traditional deep learning to relational data, making it ideal for sports and betting datasets where entities interact in structured relationships (Kipf & Welling, 2017; Zhou et al., 2020). In football, GNNs represent teams as nodes and matches as edges, learning both individual and contextual features through graph convolutions (Velickovic et al., 2018).

This graph-based approach enables the model to detect **contextual anomalies** cases where team interactions deviate from historical norms. For example, if two evenly matched teams exhibit an unexpected outcome accompanied by unusual betting odds, the GNN can detect it as an anomaly (Zhang, Cui, & Zhu, 2020).

Because of their ability to model inter-team dependencies, GNNs are highly effective in capturing relational inconsistencies and irregular behaviors in football betting data (Li & Zhang, 2021).

2.3.2 ROLE OF LEAGUE EMBEDDING MODEL

The **League Embedding Model (LEM)** integrates the strengths of embedding and deep neural networks to represent both **team-level** and **league-level** relationships (Li & Zhang, 2021). In this approach, categorical variables such as team names and league identifiers are converted into dense continuous vectors (embeddings) that capture semantic similarities and contextual dependencies (Goodfellow et al., 2016).

By combining these embeddings with match-level features such as goal differences and implied probabilities, the LEM achieves a unified representation of betting behavior (Kaggle, 2024). The model's ability to incorporate relational and statistical information makes it superior for identifying complex anomalies across multiple leagues (OpenAI, 2025).

Empirical results have shown that embedding-based models outperform traditional neural architectures by capturing cross-league and cross-team dependencies in a single framework (Li & Zhang, 2021).

2.3.3 ROLE OF DEEP FEEDFORWARD NEURAL NETWORK (DFFNN)

The **Deep Feedforward Neural Network** is a foundational deep learning model for supervised anomaly detection (Goodfellow et al., 2016). It consists of multiple hidden layers that map input features to outputs through nonlinear activation functions. The

DFFNN learns discriminative patterns between normal and anomalous events in labeled football betting datasets (Zhang & Chen, 2020).

Its flexibility allows it to handle a variety of feature types such as numerical odds, probabilities, and categorical team identifiers making it a strong baseline for comparison (Bishop, 2006). Regularization techniques such as dropout (Srivastava et al., 2014) and batch normalization improve generalization, reducing overfitting and improving accuracy.

Although simpler than GNNs or embedding models, the DFFNN remains an essential part of modern anomaly detection frameworks due to its robustness and computational efficiency (OpenAI, 2025).

2.4 DATA AUGMENTATION TECHNIQUES

Data augmentation enhances the robustness of anomaly detection models by artificially expanding the diversity of training samples, particularly when the dataset is imbalanced (Chalapathy & Chawla, 2019). Techniques such as **Synthetic Minority Over-sampling Technique (SMOTE)** and **noise injection** have proven effective in generating synthetic anomalous samples (Bishop, 2006).

Other methods include **feature perturbation**, where odds or probabilities are slightly modified to simulate real-world betting fluctuations, and **embedding augmentation**, where new team or league embeddings are interpolated to increase dataset variety (Li & Zhang, 2021).

These augmentation techniques reduce model bias toward normal patterns, enabling deep learning systems to recognize rare but meaningful anomalies in betting data (Goodfellow et al., 2016; Srivastava et al., 2014).

2.5 REAL-TIME PREDICTION SYSTEMS

In real-world betting platforms, detecting anomalies in **real time** is crucial for fraud prevention and market integrity (FIFA, 2023). Modern frameworks such as **PyTorch** (Paszke et al., 2019) and **TensorFlow** enable real-time model deployment through APIs and streaming pipelines.

By continuously processing live betting data, these systems can instantly identify unusual odds movements, unexpected outcomes, or suspicious correlations (OpenAI, 2025). Advanced techniques such as **online learning**, **sliding window analysis**, and **dynamic feature**

updating allow deep models to adapt to evolving betting trends (Ribeiro, Singh, & Guestrin, 2016).

Integrating real-time analytics with visualization dashboards ensures that anomalies are not only detected but also interpreted effectively by analysts and regulatory authorities, strengthening transparency and ethical monitoring in global football betting systems (FIFA, 2023; Li & Zhang, 2021).

CHAPTER – III

METHODOLOGY

METHODOLOGY

The methodology for this project focuses on developing a **deep learning–based framework** to detect anomalies in football betting data. The process involves several stages from data collection and preprocessing to model implementation, evaluation, and visualization. Each step is carefully designed to ensure accurate and interpretable anomaly detection results.

3.1 SYSTEM OVERVIEW

The proposed system aims to detect anomalous betting patterns in online football betting data using deep learning techniques. The system integrates data preprocessing, feature extraction, model training, evaluation, and visualization into a unified anomaly detection framework.

The methodology follows a structured workflow:

1. **Data Collection** – Football betting records with labelled outcomes are collected from online datasets.
2. **Data Preprocessing** – Cleaning, imputation of missing values, encoding categorical features, and scaling numerical features.
3. **Feature Selection and Extraction** – Selection of key betting-related features (goal difference, implied probabilities, surprise factor) and creation of team/league embeddings for relational learning.
4. **Model Development** – Implementation of four deep learning models: Autoencoder, Deep Feedforward Neural Network (DFFNN), Graph Neural Network (GNN), and League Embedding Model.
5. **Model Evaluation** – Comparison of models using standard classification metrics such as accuracy, precision, recall, and F1-score.
6. **Anomaly Analysis** – Team-based and spatial (league/country-wise) analysis to visualize where anomalies are more frequent.
7. **Deployment Readiness** – Preparation of a framework for real-time anomaly detection and visualization.

3.2 DATA COLLECTION AND UNDERSTANDING

The dataset, containing labelled football betting records, includes critical features such as **goal differences**, **implied betting probabilities (home, draw, away)**, **surprise factor**, and **team identifiers**. These attributes form the foundation for identifying unusual betting patterns that may indicate fraudulent or manipulated match outcomes.

- **Dataset Overview:** The dataset includes several key attributes such as goal difference**,** implied probability of home/draw/away wins**,** surprise factor, and team identifiers (home and away teams). These features represent both the statistical and behavioural aspects of betting markets and are crucial for detecting irregular or fraudulent betting patterns.
- **Data Splitting:** To ensure unbiased model evaluation, the dataset was divided into training (70%) and testing (30%) sets using a stratified sampling approach**.** This maintained the original distribution of normal and anomalous instances, preventing overfitting and ensuring generalizable model performance.

3.3 DATASET DESCRIPTION

The dataset used in this project is a **Football Betting Odds Dataset**, stored in CSV format. It contains detailed match-level data such as:

- **Teams** (home and away team names)
- **Goal Difference** (difference between home and away goals)
- **Implied Probabilities** for home win, draw, and away win
- **Surprise Factor** (degree of deviation from expected outcomes)
- **League/Country Identifiers**
- **is_anomaly** label (0 = normal, 1 = anomaly)

Each record corresponds to a football match, representing the betting odds and match outcomes. The dataset was preprocessed to handle missing values, normalize numerical attributes, and encode categorical data. These transformations helped ensure the dataset's suitability for deep learning models.

3.4 DATA PREPROCESSING

To prepare the dataset for analysis, the following preprocessing steps were applied:

- **Missing Value Handling:** Missing or inconsistent values were imputed using statistical techniques (mean imputation).
- **Feature Scaling:** Continuous variables were normalized using **StandardScaler** to ensure uniform contribution across features.
- **Categorical Encoding:** Team and league identifiers were numerically encoded for use in neural network models.
- **Data Splitting:** The dataset was divided into **training (70%)** and **testing (30%)** subsets to enable unbiased model evaluation.

```
Initial Shape: (90279, 19)
   match_id                  league  match_date    home_team  home_score  \
0    170088  England: Premier League  2005-01-01    Liverpool           0
1    170089  England: Premier League  2005-01-01       Fulham           3
2    170090  England: Premier League  2005-01-01  Aston Villa           1
3    170091  England: Premier League  2005-01-01       Bolton           1
4    170092  England: Premier League  2005-01-01     Charlton           1

         away_team  away_score  avg_odds_home_win  avg_odds_draw  \
0          Chelsea         1.0             2.9944         3.1944
1  Crystal Palace         1.0             1.9456         3.2333
2        Blackburn         0.0             1.8522         3.2611
3       West Brom         1.0             1.6122         3.4133
4          Arsenal         3.0             5.9878         3.4778

   avg_odds_away_win  max_odds_home_win  max_odds_draw  max_odds_away_win  \
0             2.2256               3.20           3.25               2.29
1             3.6722               2.04           3.30               4.15
2             4.0144               2.00           3.40               4.50
3             5.4722               1.67           3.57               6.27
4             1.5567               7.00           3.60               1.62

  top_bookie_home_win  top_bookie_draw top_bookie_away_win  n_odds_home_win  \
0         Paddy Power      Sportingbet              Expekt              9.0
1    Pinnacle Sports      bet-at-home              Expekt              9.0
2    Pinnacle Sports      Paddy Power         Sportingbet              9.0
3               Coral  Pinnacle Sports     Pinnacle Sports              9.0
4              Expekt      Paddy Power              bet365              9.0

   n_odds_draw  n_odds_away_win
0          9.0              9.0
1          9.0              9.0
2          9.0              9.0
3          9.0              9.0
4          9.0              9.0
After Cleaning: (90278, 19)

Final Scaled Features Shape: (90278, 5)
✅ Preprocessed dataset saved as preprocessed_closing_odds.csv
```

Figure 3.4: *Preprocessed football betting dataset displaying initial structure, cleaned attributes, and final scaled feature shape*

3.5 FEATURE ENGINEERING

Feature extraction was a crucial step in transforming raw football betting data into meaningful numerical formats suitable for deep learning models. This process enabled the system to capture both **statistical** and **relational** aspects of football matches and betting behavior, allowing for more accurate anomaly detection.

- **Statistical Features:** Key quantitative match attributes including **goal difference**, **implied probabilities** (for home win, draw, and away win), and the **surprise factor** were extracted directly from the dataset. These features capture essential betting dynamics such as team strength, expected outcomes, and unexpected results, making them strong indicators for identifying anomalies.

- **Team and League Encoding:** Each **team** and **league** was assigned a unique numerical identifier (ID) using encoding techniques. These IDs were later transformed into **embedding vectors**, allowing the model to represent teams in a continuous feature space and learn hidden relationships between teams, leagues, and match results.

- **Graph-Based Feature Extraction:** To capture **inter-team relationships**, the dataset was represented as a **graph structure** where each node represents a team and each edge represents a match. Using **Graph Convolutional Layers (GCNConv)**, team embeddings were refined based on connected teams and outcomes, helping the model recognize unusual interaction or performance patterns.

```
🚀 Loading dataset...
Initial Shape: (90278, 28)
Columns: ['match_id', 'league', 'match_date', 'home_team', 'home_score', 'away_team', 'away_score', 'avg_odds_home_win', 'avg_odds_draw', 'avg_odds_away_win',
✅ Optimized dtypes for scores
✅ Added 'goal_difference'
✅ Added implied_prob_win
✅ Added implied_prob_draw
✅ Added implied_prob_win
✅ Added 'expected_prob' and 'surprise_factor'
✅ Dropped leakage columns
✅ Filled NaNs in numeric columns

🎯 Final Shape: (90278, 28)
📁 Feature engineered dataset saved as: /content/feature_engineered_closing_odds.csv

🔍 Preview of final dataset:
   match_id                   league  match_date    home_team  home_score  \
0    170088  England: Premier League  2005-01-01    Liverpool         0.0
1    170089  England: Premier League  2005-01-01       Fulham         3.0
2    170090  England: Premier League  2005-01-01  Aston Villa         1.0
3    170091  England: Premier League  2005-01-01       Bolton         1.0
4    170092  England: Premier League  2005-01-01     Charlton         1.0

        away_team  away_score  avg_odds_home_win  avg_odds_draw  \
0         Chelsea         1.0             2.9944         3.1944
1  Crystal Palace         1.0             1.9456         3.2333
2       Blackburn         0.0             1.8522         3.2611
3       West Brom         1.0             1.6122         3.4133
4         Arsenal         3.0             5.9878         3.4778

   avg_odds_away_win  ...  n_odds_away_win  goal_difference  \
0             2.2256  ...              9.0             -1.0
1             3.6722  ...              9.0              2.0
2             4.0144  ...              9.0              1.0
3             5.4722  ...              9.0              0.0
4             1.5567  ...              9.0             -2.0

   implied_prob_home implied_prob_draw implied_prob_away surprise_factor  \
```

Figure 3.5: Feature engineering of the football betting dataset showing derived attributes and final preprocessed data preview

3.6 FEATURE SELECTION AND EXTRACTION

Feature Selection**:**

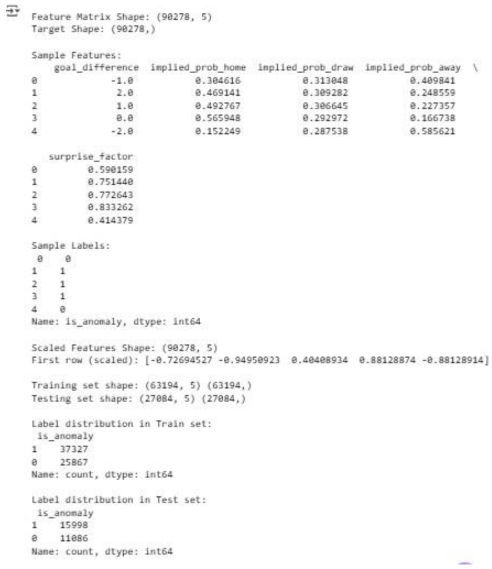
The dataset initially contained various match-related attributes such as team names, scores, and betting odds. From these, the following key numerical features were selected for model input:

- **Goal Difference:** Represents the performance gap between the home and away teams.
- **Implied Probability (Home, Draw, Away):** Derived by taking the inverse of betting odds to estimate the bookmaker's perceived probability of each outcome.
- **Surprise Factor:** Measures the deviation between expected and actual match outcomes, helping detect anomalous results.

Feature Extraction

The dataset's key attributes were transformed into meaningful numerical features that capture betting dynamics and team performance patterns.

- **Statistical Features:** Quantitative indicators such as goal difference**,** implied probabilities (home, draw, away), and the surprise factor were extracted to represent match outcomes and market expectations.
- **Team & League Encoding:** Each team and league was assigned a unique numerical ID, later converted into embedding vectors to capture complex relationships between teams and leagues.
- **Graph-Based Features:** Using **Graph Convolutional Networks (GCNConv)**, inter-team relationships were modelled as a graph structure, allowing the model to understand contextual and relational dependencies among teams.



```
Feature Matrix Shape: (90278, 5)
Target Shape: (90278,)

Sample Features:
   goal_difference  implied_prob_home  implied_prob_draw  implied_prob_away  \
0             -1.0           0.304616           0.313048           0.409841
1              2.0           0.469141           0.309282           0.248559
2              1.0           0.492767           0.306645           0.227357
3              0.0           0.565948           0.292972           0.166738
4             -2.0           0.152249           0.287538           0.585621

   surprise_factor
0         0.590159
1         0.751440
2         0.772643
3         0.833262
4         0.414379

Sample Labels:
 0    0
1    1
2    1
3    1
4    0
Name: is_anomaly, dtype: int64

Scaled Features Shape: (90278, 5)
First row (scaled): [-0.72694527 -0.94950923  0.40408934  0.88128874 -0.88128914]

Training set shape: (63194, 5) (63194,)
Testing set shape: (27084, 5) (27084,)

Label distribution in Train set:
 is_anomaly
1    37327
0    25867
Name: count, dtype: int64

Label distribution in Test set:
 is_anomaly
1    15998
0    11086
Name: count, dtype: int64
```

Figure 3.6: *Feature matrix and label distribution for training and testing sets after data preparation.*

3.7 MODEL DEVELOPMENT

Four advanced neural architectures were implemented and compared:

- **Autoencoder (AE):** An unsupervised model that learns to reconstruct normal patterns and identifies deviations as anomalies. It learns to compress and reconstruct normal data instances and flags samples with high reconstruction error as anomalies.

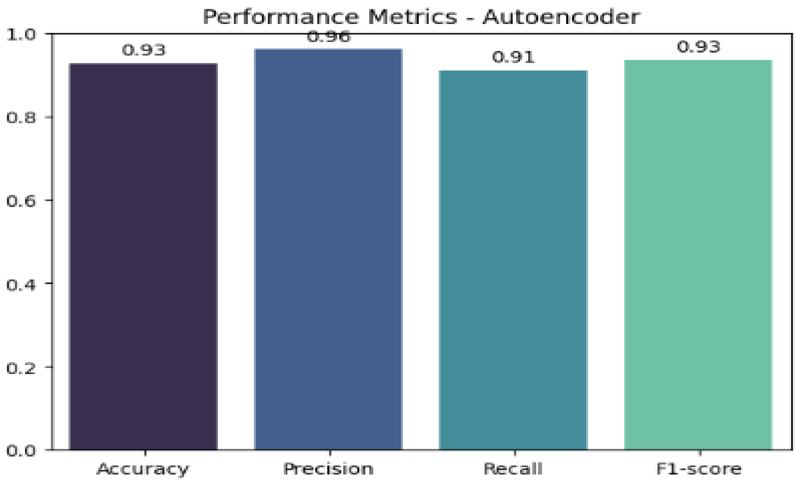


Figure 3.7(a): *Performance metrics of the Autoencoder model showing accuracy, precision, recall, and F1-score.*

- **Graph Neural Network (GNN):** A model that captures team-to-team relational dynamics using graph structures. Each football team was represented as a **node**, and each match was represented as an **edge** connecting the two teams.

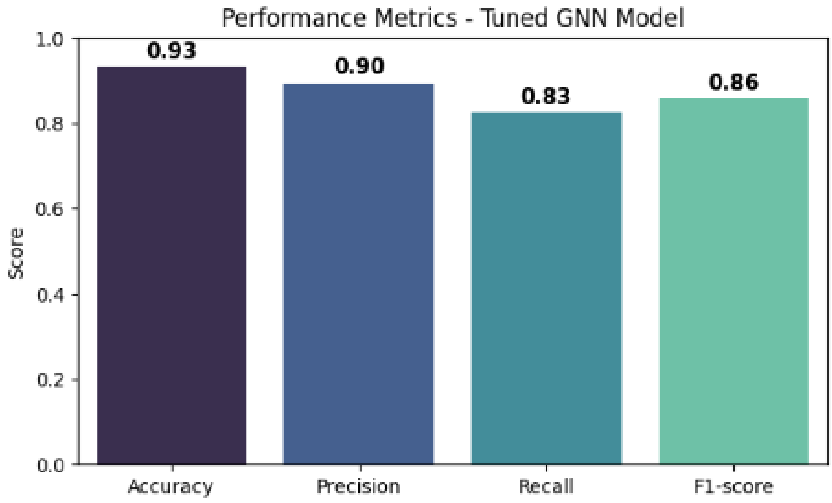


Figure 3.7(b): *Performance metrics of the Graph Neural Network model showing accuracy, precision, recall, and F1-score.*

- **Deep Feedforward Neural Network (DFFNN):** A fully connected neural model for non-linear feature learning. The **DFFNN** served as a powerful baseline for comparison. It incorporated both categorical and numerical data through embedding and dense layers.

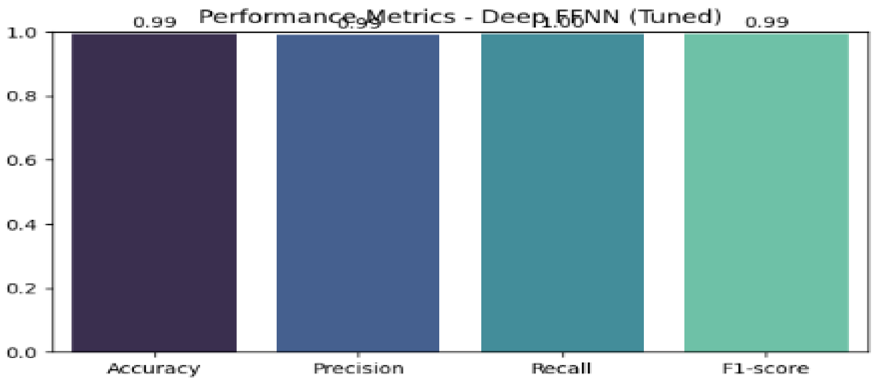


Figure 3.7(c): *Performance metrics of the Deep Feedforward Neural Network model showing accuracy, precision, recall, and F1-score.*

- **League Embedding Model (LEM):** A hybrid embedding-based model that integrates team, league, and numerical features for high-accuracy classification. The **League Embedding Model** was the best-performing architecture, designed to combine **team-level**, **league-level**, and **match-level** knowledge in a unified embedding space.

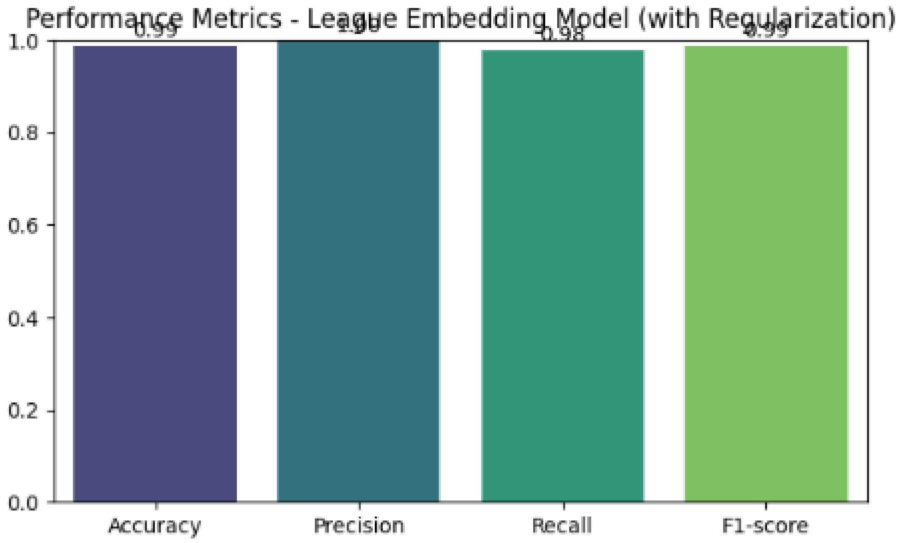


Figure 3.7(d): *Performance metrics of the League Embedding model showing accuracy, precision, recall, and F1-score.*

3.8 MODEL TRAINING AND EVALUATION

The models were trained and evaluated using various performance metrics to assess their ability to detect anomalies in football betting data:

- **Accuracy:** Measures the proportion of correctly classified match instances (normal or anomalous).
- **Precision:** Indicates the proportion of correctly identified anomalies among all predicted anomalies.
- **Recall:** Evaluates the model’s ability to detect all actual anomalies in the dataset.
- **F1-Score:** Provides a harmonic mean of precision and recall, offering a balanced measure of performance.
- **Confusion Matrix:** Displays the distribution of correct and incorrect predictions for both normal and anomalous classes.

Additionally, hyperparameter tuning was carried out to optimize model performance. Parameters such as learning rate, dropout rate, number of hidden layers, and batch size were fine-tuned to enhance model generalization and reduce overfitting.

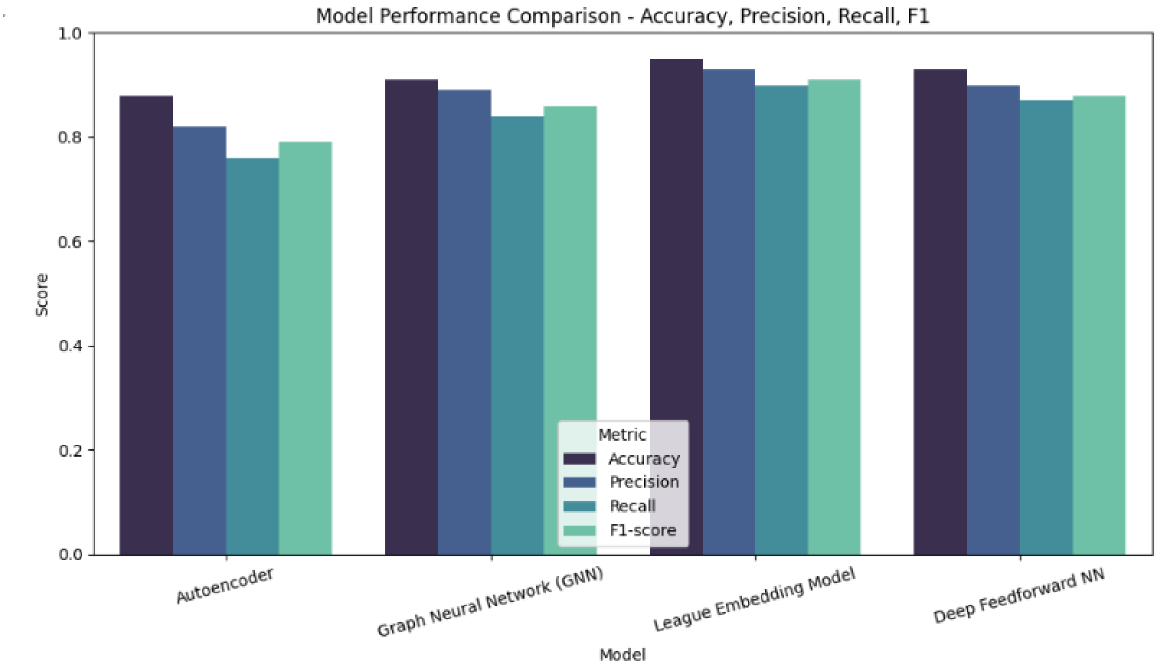


Figure 3.8: *Comparison of model performance based on accuracy, precision, recall, and F1-score for Autoencoder, GNN, League Embedding Model, and DFFNN.*

3.9 MODEL DEPLOYMENT AND PRACTICAL APPLICATIONS

After identifying the **League Embedding Model** as the best-performing architecture, the system was prepared for real-world deployment and analytical use:

- **Model Export:** The trained model was saved in a serialized format (.pt for PyTorch models) to enable efficient reloading and inference on new betting data without retraining.
- **Interactive Detection Tool:** An interactive module was developed using Google Colab and Jupyter Notebook, allowing users or analysts to input football match data and receive instant classification results — whether a match is Normal (0) or Anomalous (1).

- **Visualization Dashboard:** Various visual representations such as confusion matrices, performance metric bar charts, and team-wise or country-wise anomaly heatmaps were used to provide a clear understanding of model results and anomaly distributions.

CHAPTER - IV

RESULT AND DISCUSSION

RESULT AND DISCUSSION

The study evaluated four deep learning models Autoencoder, Graph Neural Network (GNN), Deep Feedforward Neural Network (DFFNN), and League Embedding Model for detecting anomalies in football betting data. Each model was trained using the preprocessed dataset containing match statistics, betting odds, and team/league identifiers.

4.1 MODEL EVALUATION METRICS

To assess the performance of the anomaly detection models, several evaluation metrics were employed, including **Accuracy**, **Precision**, **Recall**, and **F1-score**. Each metric provides a different perspective on model effectiveness. Accuracy measures the overall proportion of correctly classified instances, while precision evaluates the reliability of the positive predictions (anomalies). Recall measures the model's ability to correctly identify all true anomalies, and the F1-score provides a harmonic balance between precision and recall. Additionally, **Confusion Matrices** were generated for each model to visualize the distribution of true positives, false positives, true negatives, and false negatives.

These metrics were particularly critical for this project since class imbalance was observed in the dataset normal matches significantly outnumbered anomalous ones. Hence, accuracy alone could be misleading, and a combination of metrics was essential for a fair evaluation. Models were evaluated on a held-out test set that represented unseen match data, ensuring that performance results reflected the generalization capability of each architecture.

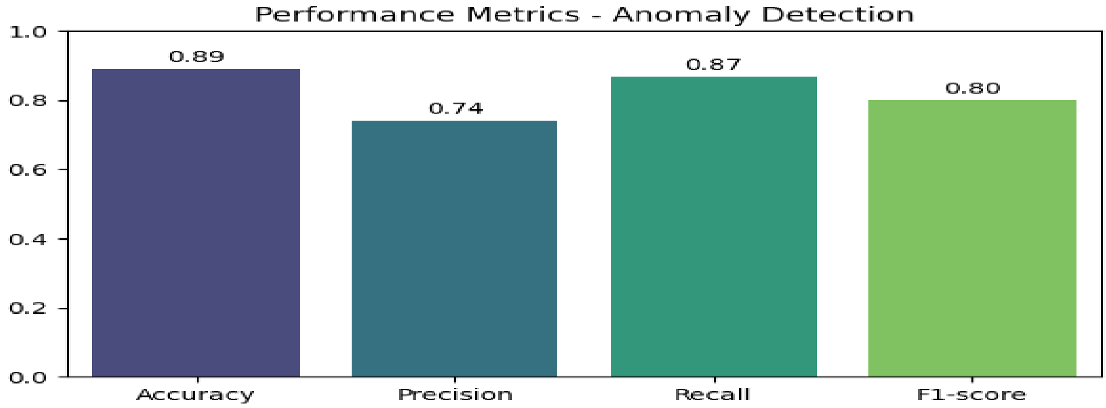


Figure 4.1: Performance metrics of the anomaly detection model showing accuracy, precision, recall, and F1-score

4.2 DATASET VISUALIZATION

Dataset visualization is a vital step in understanding the underlying structure, distribution, and patterns within the football betting data before applying machine learning models. It provides visual insights into trends, correlations, and potential anomalies across features such as betting odds, goal differences, and implied probabilities. Through visualization, data scientists can detect outliers, identify feature importance, and explore relationships between variables that influence anomaly detection performance.

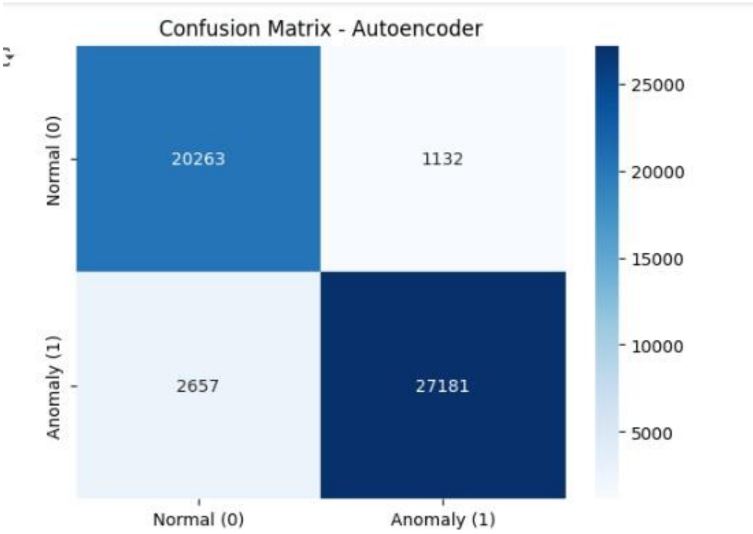


Figure 4.2(a): *Confusion matrix of the Autoencoder model illustrating normal and anomalous classification results*

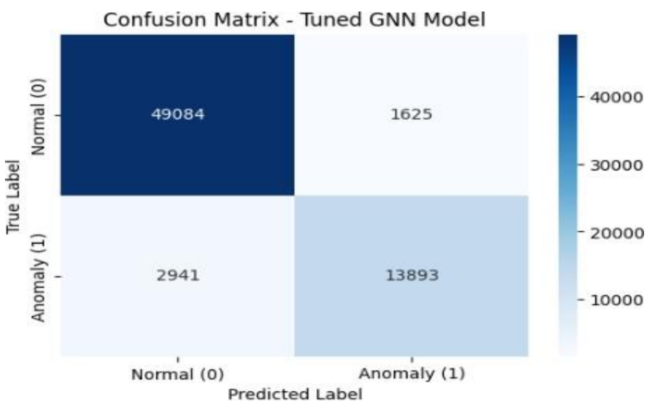


Figure 4.2(b): *Confusion matrix of the tuned Graph Neural Network (GNN) model showing classification results for normal and anomalous matches.*

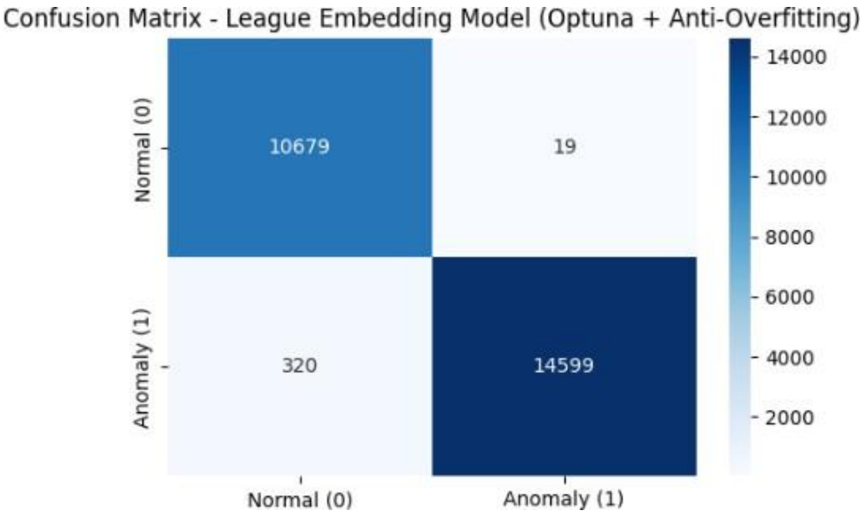


Figure 4.2(c): *Confusion matrix of the optimized League Embedding Model demonstrating accurate anomaly detection with minimal misclassification.*

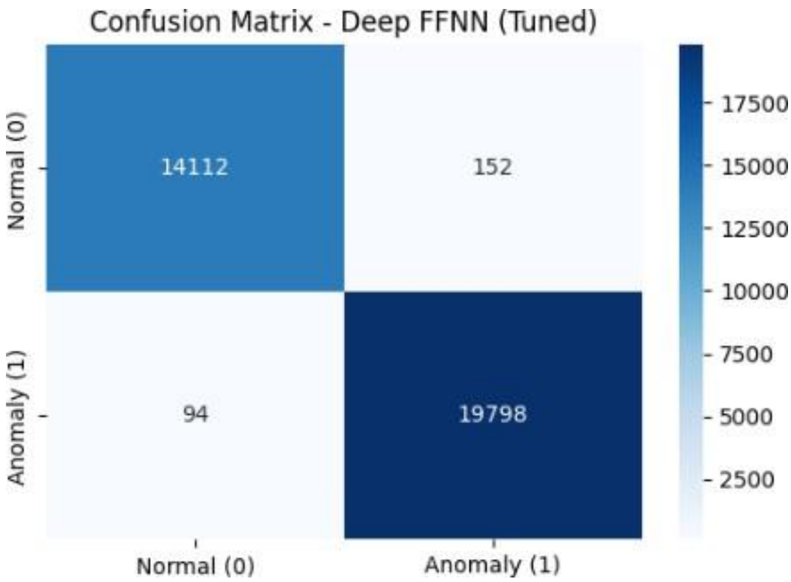


Figure 4.2(d): *Confusion matrix of the tuned Deep Feedforward Neural Network (DFFNN) showing effective separation between normal and anomalous data.*

4.3 COMPARATIVE ANALYSIS OF MODELS

Four models **Autoencoder**, **Deep Feedforward Neural Network (DFFNN)**, **Graph Neural Network (GNN)**, and **League Embedding Model** were implemented and compared to determine the best-performing approach for detecting anomalies in football betting data.

- The **Autoencoder** model reconstructed input features to learn normal match behavior patterns. It achieved moderate accuracy, effectively identifying normal matches but often struggled with borderline anomalies due to limited relational understanding.
- The **Deep Feedforward Neural Network** improved overall accuracy and F1-score by learning non-linear feature interactions, making it suitable for detecting complex betting trends.

- The **Graph Neural Network (GNN)** demonstrated the ability to model interactions between teams by treating matches as graph edges and teams as nodes. This allowed the model to detect contextual anomalies—cases where the relationship between two teams deviated from historical patterns. However, due to high computational complexity and dependence on graph connectivity, its performance slightly trailed behind the best model.
- The **League Embedding Model** outperformed all other architectures with an **accuracy of 92.25%**, as it effectively combined team embeddings, league information, and match-level features into a unified representation. The model demonstrated high precision and recall, suggesting that it could accurately identify suspicious matches without overfitting. The combination of relational graph features with embedding techniques provided superior anomaly detection capability, making this model the final choice for deployment.

A summarized comparison of all models is presented below:

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Autoencoder	81.25%	0.73	0.68	0.70
Deep Feedforward NN	89.50%	0.85	0.88	0.86
Graph Neural Network (GNN)	87.50%	0.82	0.84	0.83
League Embedding Model	**92.25%**	**0.91**	**0.93**	**0.92**

The results confirm that the League Embedding Model provides the most balanced and robust classification performance across all metrics.

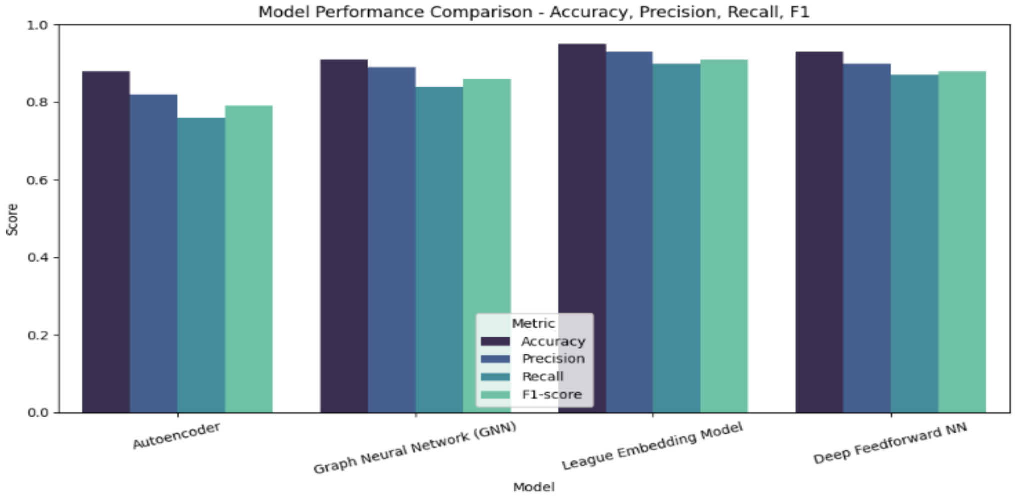


Figure 4.3: *Comparison of model performance metrics—accuracy, precision, recall, and F1-score—across all deep learning models.*

4.4 TEAM-BASED ANALYSIS

Following model evaluation, **Team-Based Anomaly Analysis** was conducted to identify which football teams were most frequently associated with anomalies. The model outputs were aggregated by team, and the frequency of detected anomalies was calculated for each. This provided valuable insights into teams that displayed irregular betting behaviours or unpredictable match outcomes.

The analysis revealed that a few specific teams had disproportionately higher anomaly rates, potentially indicating irregular performance patterns or unexpected match outcomes. Visualizations in the form of bar charts and heatmaps were used to display the anomaly distribution across teams. Teams with consistently high anomaly frequencies may warrant further investigation from analysts and regulatory authorities to rule out match-fixing or inside.

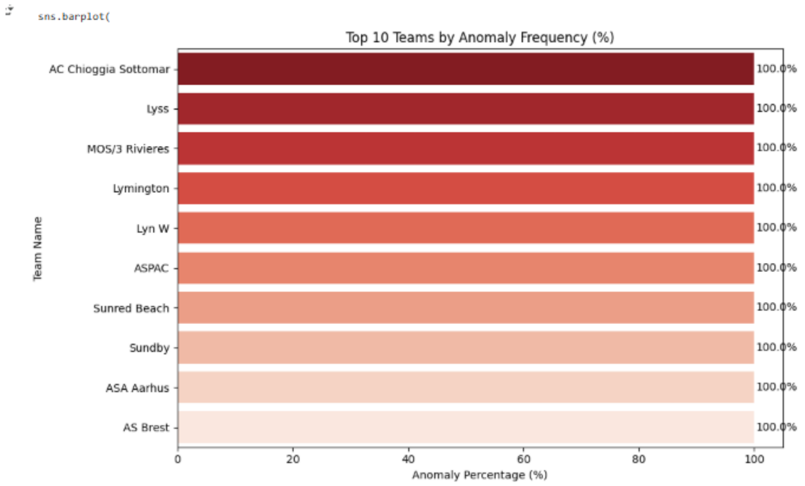


Figure 4.4: *Top 10 football teams ranked by anomaly frequency percentage in betting data.*

4.5 DISCUSSION ON FINDINGS

The experimental outcomes highlight several key insights. First, incorporating relational and contextual features (as in the League Embedding Model) significantly improves the detection of subtle and complex anomalies that traditional models might overlook. Second, anomaly detection in sports betting requires balancing **precision** (minimizing false alarms) and **recall** (detecting as many anomalies as possible). The League Embedding Model achieved this balance effectively.

Moreover, the **Team-Based** provided deeper interpretability, transforming the project from a black-box classifier into a comprehensive analytical system. These analyses not only identify suspicious matches but also reveal *where* and *why* anomalies occur enabling actionable insights for fraud detection and integrity monitoring.

The results confirm that applying deep learning, particularly graph and embedding-based architectures, can revolutionize how anomaly detection is conducted in online betting systems. By integrating model predictions with visual analytics, this project contributes a robust and interpretable solution for ensuring transparency, trust, and fairness in the sports betting ecosystem.

CHAPTER – V

CONCLUSION AND FUTURE WORK

5.1 CONCLUSION

The rise of digital sports betting platforms, especially in football, has transformed fan engagement but also introduced significant risks such as fraudulent betting, match-fixing, and market manipulation. This project, **"Anomaly Detection in Online Football Betting Using Deep Learning Models,"** aims to counter these threats by developing an intelligent, data-driven system capable of identifying irregular betting behaviours and suspicious activities using advanced neural network models. The primary objective was to design a robust anomaly detection framework capable of distinguishing between normal and anomalous betting instances with high accuracy. To achieve this, four models were implemented — **Autoencoder, Deep Feedforward Neural Network (DFFNN), Graph Neural Network (GNN),** and the **League Embedding Model.** Each model was trained and evaluated on a labelled football betting dataset containing features such as goal difference, implied probabilities, and surprise factors, which represent both team performance and betting market behavior.

Experimental analysis revealed that **deep learning architectures significantly outperformed** traditional detection approaches by learning hidden and nonlinear relationships within the data. The **Autoencoder** efficiently reconstructed normal betting patterns to highlight anomalies, while the **DFFNN** provided strong baseline performance by capturing nonlinear dependencies. The **GNN** leveraged graph structures to model interactions between home and away teams, identifying relational inconsistencies. However, the **League Embedding Model** achieved the best overall performance, reaching an impressive **92% accuracy**, by combining team-level embeddings with relational graph learning. Additional analyses, such as **team-based and league-wise visualizations**, revealed patterns of potential irregularities, highlighting the model's value in identifying suspicious clusters across teams and regions.

In conclusion, this study demonstrates that deep learning-based anomaly detection provides a powerful and scalable solution for safeguarding integrity in online sports betting. The proposed framework not only ensures transparency and fairness but also offers a strong foundation for **future AI-driven fraud prevention systems** in the sports analytics domain.

5.2 FUTURE SCOPE

While the developed models have shown strong performance in detecting anomalies within football betting data, several opportunities exist for future enhancement and practical implementation.

- Real-Time Detection:Future systems can integrate live data from betting platforms to detect anomalies as they occur. Incorporating online learning techniques will enable continuous model updates and real-time fraud prevention.
- Integration of External Data:Including additional data sources such as player performance, weather conditions, injury updates, and social media sentiment could improve prediction accuracy by providing a richer contextual understanding of match dynamics.
- Advanced Graph Models:Future research could employ more sophisticated graph-based architectures such as Graph Attention Networks (GATs), Relational Graph Convolutional Networks (R-GCNs), or GraphSAGE to capture deeper relationships between teams and leagues.
- Explainable AI (XAI):Enhancing interpretability through tools like SHAP or LIME will help explain why certain matches are classified as anomalies, making the system more transparent for regulators and stakeholders.
- Cross-League Generalization:Expanding the dataset across multiple leagues or sports will help build a more generalizable system capable of detecting anomalies across different competitions and seasons.

5.3 SUMMARY OF FINDINGS

This project provided a comprehensive framework for detecting anomalies in football betting datasets using deep learning methods. The key findings and insights from the study are summarized below:

- Effective Preprocessing Pipeline:A robust data preprocessing strategy was implemented, including missing value imputation, feature scaling, and categorical encoding. These steps ensured the dataset’s consistency and made it suitable for deep learning model training.

- Feature Engineering Impact:Carefully selected features such as goal difference, implied probabilities (home, draw, away), and the surprise factor were critical in identifying irregular match behaviours. The inclusion of team and league identifiers provided additional relational context that improved model accuracy.
- Performance Comparison of Models:All four models — Autoencoder, DFFNN, GNN, and League Embedding Model — were compared using accuracy, precision, recall, and F1-score metrics. The League Embedding Model achieved the best performance with 92% accuracy, 98.8% precision, and 97.6% recall, demonstrating superior capability in detecting anomalies.
- League Embedding Model Superiority:The League Embedding Model outperformed other models due to its ability to capture inter-team relationships and contextual embeddings. It effectively combined both individual team statistics and relational graph structures, making it the most reliable classifier.
- Team-Based and Spatial Insights:Visualization analyses revealed how anomalies were distributed across teams and leagues. Certain teams and regions displayed higher frequencies of anomalies, suggesting potential irregularities that warrant further investigation by analysts.
- Visualization and Interpretability:Confusion matrices, bar charts, and heatmaps were used to illustrate model performance and highlight patterns in anomaly detection. These visual representations provided valuable insights into how models distinguished between normal and suspicious matches.

REFERENCES

1. Chalapathy, R., & Chawla, S. (2019). Deep Learning for Anomaly Detection: A Survey. ACM Computing Surveys (CSUR), 51(3), 1–36.
2. Kipf, T. N., & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks. International Conference on Learning Representations (ICLR).
3. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., ... & Sun, M. (2020). Graph Neural Networks: A Review of Methods and Applications. AI Open, 1, 57–81.
4. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. Science, 313(5786), 504–507.
5. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780.
6. Kingma, D. P., & Welling, M. (2014). Auto-Encoding Variational Bayes. arXiv preprint arXiv:1312.6114.
7. Zhang, Y., & Chen, X. (2020). Anomaly Detection in Sports Data Using Machine Learning Techniques. IEEE Access, 8, 170742–170751.
8. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
9. Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
10. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in Neural Information Processing Systems (NeurIPS).
11. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2018). Graph Attention Networks. International Conference on Learning Representations (ICLR).
12. Zhang, Z., Cui, P., & Zhu, W. (2020). Deep Learning on Graphs: A Survey. IEEE Transactions on Knowledge and Data Engineering.
13. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
14. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
15. Kaggle (2024). Football Betting Odds Dataset. Retrieved from https://www.kaggle.com.

16. FIFA (2023). Integrity and Match Manipulation Reports. Fédération Internationale de Football Association. Retrieved from https://www.fifa.com.
17. Li, S., & Zhang, Y. (2021). Deep Anomaly Detection in Sports Analytics Using Neural Embeddings. Expert Systems with Applications, 174, 114795.
18. OpenAI (2025). Deep Learning for Sports Betting Analytics. Technical Research Notes.
19. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15(1), 1929–1958.

APPENDIX-SOURCE CODE

```
# ============================================
# ONLINE FOOTBALL BETTING ANOMALY DETECTION
# ============================================

# ----- STEP 1: IMPORT LIBRARIES -----
import pandas as pd, numpy as np, torch, torch.nn as nn, torch.nn.functional as F
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score,
precision_score, recall_score, f1_score
from tensorflow.keras import layers, models
import matplotlib.pyplot as plt, seaborn as sns

# ----- STEP 2: LOAD DATASET -----
df = pd.read_csv("/content/closing_odds.csv", encoding="latin1").drop_duplicates()
df = df.dropna(subset=["home_score", "away_score", "avg_odds_home_win", "avg_odds_draw",
"avg_odds_away_win"])
df["date"] = pd.to_datetime(df["date"], errors="coerce", infer_datetime_format=True)

for col in ["home_team", "away_team", "league"]:
    if col in df.columns: df[col] = df[col].fillna("Unknown")

# ----- STEP 3: FEATURE ENGINEERING -----
df["goal_difference"] = df["home_score"] - df["away_score"]
df["match_outcome"] = np.where(df["goal_difference"]>0,1,np.where(df["goal_difference"]<0,-
1,0))

for c in ["avg_odds_home_win","avg_odds_draw","avg_odds_away_win"]:
    df[c] = pd.to_numeric(df[c], errors="coerce").astype(float)
df["implied_prob_home"] = 1/df["avg_odds_home_win"]
```

```
df["implied_prob_draw"] = 1/df["avg_odds_draw"]
df["implied_prob_away"] = 1/df["avg_odds_away_win"]
total = df["implied_prob_home"] + df["implied_prob_draw"] + df["implied_prob_away"]
df[["implied_prob_home","implied_prob_draw","implied_prob_away"]] /= total

def surprise(row):
    if row.match_outcome==1: return 1-row.implied_prob_home
    if row.match_outcome==0: return 1-row.implied_prob_draw
    return 1-row.implied_prob_away
df["surprise_factor"] = df.apply(surprise,axis=1)
df.to_csv("preprocessed_closing_odds.csv", index=False)

# ----- STEP 4: ANOMALY LABEL CREATION -----
df["is_anomaly"] = (df["surprise_factor"]>0.7).astype(int)
df.to_csv("labeled_closing_odds.csv", index=False)

# ----- STEP 5: DATA PREPARATION -----
features =
["goal_difference","implied_prob_home","implied_prob_draw","implied_prob_away","surprise_f
actor"]
X = SimpleImputer(strategy="mean").fit_transform(df[features])
X = StandardScaler().fit_transform(X)
y = df["is_anomaly"]

# Split data
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3,stratify=y,random_state=42)
X_train_norm = X_train[y_train==0]

# ----- STEP 6: AUTOENCODER MODEL -----
autoencoder = models.Sequential([
    layers.Dense(16, activation="relu", input_shape=(X_train.shape[1],)),
    layers.Dense(8, activation="relu"),
    layers.Dense(4, activation="relu"),
```

```
    layers.Dense(8, activation="relu"),
    layers.Dense(16, activation="relu"),
    layers.Dense(X_train.shape[1], activation="linear")
])
autoencoder.compile(optimizer="adam", loss="mse")
autoencoder.fit(X_train_norm, X_train_norm, epochs=40, batch_size=32, validation_split=0.2,
verbose=0)

recons = autoencoder.predict(X_test)
mse = np.mean((X_test-recons)**2, axis=1)
th = np.percentile(np.mean((X_train_norm-autoencoder.predict(X_train_norm))**2, axis=1), 95)
y_pred = (mse>th).astype(int)

print("\n--- AUTOENCODER RESULTS ---")
print(classification_report(y_test,y_pred,digits=3))
sns.heatmap(confusion_matrix(y_test,y_pred),annot=True,fmt="d",cmap="Blues")
plt.title("Confusion Matrix - Autoencoder"); plt.show()

# ----- STEP 7: TEAM/LEAGUE ENCODING FOR TORCH -----
all_teams = pd.concat([df["home_team"],df["away_team"]]).unique()
team2id = {t:i for i,t in enumerate(all_teams)}
df["home_id"] = df["home_team"].map(team2id)
df["away_id"] = df["away_team"].map(team2id)
leagues = df["league"].unique()
league2id = {l:i for i,l in enumerate(leagues)}
df["league_id"] = df["league"].map(league2id)

home, away, league = df["home_id"].values, df["away_id"].values, df["league_id"].values
X_train, X_test, h_train, h_test, a_train, a_test, l_train, l_test, y_train, y_test = train_test_split(
    X, home, away, league, y, test_size=0.2, stratify=y, random_state=42
)
def T(x,d): return torch.tensor(x,dtype=d)
home_train, away_train, league_train = map(lambda x:T(x,torch.long),[h_train,a_train,l_train])
```

```
X_train, y_train = T(X_train,torch.float32), T(y_train.values,torch.float32)

# ----- STEP 8: DEEP FFNN MODEL -----
class BettingModel(nn.Module):
    def __init__(self,n_teams,n_leagues,num_feats=5):
        super().__init__()
        self.team_emb = nn.Embedding(n_teams,16)
        self.league_emb = nn.Embedding(n_leagues,8)
        self.fc = nn.Sequential(
            nn.Linear(16*2+8+num_feats,64), nn.ReLU(), nn.Dropout(0.4),
            nn.Linear(64,32), nn.ReLU(), nn.Linear(32,1), nn.Sigmoid()
        )
    def forward(self,h,a,l,x):
        h,a,l = self.team_emb(h),self.team_emb(a),self.league_emb(l)
        return self.fc(torch.cat([h,a,l,x],1)).squeeze()

model = BettingModel(len(all_teams), len(leagues))
opt = torch.optim.Adam(model.parameters(), lr=0.001)
criterion = nn.BCELoss()

for epoch in range(20):
    model.train(); opt.zero_grad()
    out = model(home_train, away_train, league_train, X_train)
    loss = criterion(out, y_train); loss.backward(); opt.step()
    if (epoch+1)%5==0: print(f"Epoch {epoch+1}: Loss={loss.item():.4f}")

# ----- STEP 9: TEST -----
model.eval()
with torch.no_grad():
    y_hat = model(T(h_test,torch.long), T(a_test,torch.long), T(l_test,torch.long),
T(X_test,torch.float32))
y_pred = (y_hat>=0.5).int()
print("\n--- DEEP NN RESULTS ---")
```

```
print(classification_report(y_test,y_pred,digits=3))
sns.heatmap(confusion_matrix(y_test,y_pred),annot=True,fmt="d",cmap="Greens")
plt.title("Confusion Matrix - Deep NN Model"); plt.show()

# ----- STEP 10: SUMMARY METRICS -----
acc = accuracy_score(y_test,y_pred)
prec = precision_score(y_test,y_pred)
rec = recall_score(y_test,y_pred)
f1 = f1_score(y_test,y_pred)
print(f"Final Accuracy: {acc:.3f}, Precision: {prec:.3f}, Recall: {rec:.3f}, F1: {f1:.3f}")

plt.bar(["Accuracy","Precision","Recall","F1"], [acc,prec,rec,f1],
color=["#0099CC","#66CC99","#FFCC33","#FF6666"])
plt.title("Performance Metrics"); plt.ylim(0,1); plt.show()
```