

ANN and CNN

August 4, 2025

```
[1]: #ECGR-4105 ANN and CNN Ai Stock Predictor
#Steve Revens
#Harrison Hall

import yfinance as yf
import pandas as pd
import numpy as np
import os
import time
from datetime import datetime
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.metrics import mean_absolute_error
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Conv1D, MaxPooling1D, Flatten,
    ↪InputLayer

# Debugging help
NUM_COMPANIES = 500                                # Set to 500 for full run
FORCE_REDOWNLOAD = True                            # Set to False if the spreadsheets are
    ↪downloaded and current, stops the program from re-downloading the
    ↪spreadsheets
SAVE_DIR = "data"
FUNDAMENTAL_DIR = "fundamentals"

# Get S&P tickers
def get_sp500_tickers():
    url = "https://datahub.io/core/s-and-p-500-companies/r/constituents.csv"
    df = pd.read_csv(url)
    return df['Symbol'].str.replace('.', '-', regex=False).tolist()

# Download the data
def download_stock_data(tickers, start="2000-01-01", end=None,
    ↪save_dir=SAVE_DIR, force_redownload=False):
```

```

if end is None:
    end = datetime.today().strftime("%Y-%m-%d")
os.makedirs(save_dir, exist_ok=True)
os.makedirs(FUNDAMENTAL_DIR, exist_ok=True)
data = {}
for i, ticker in enumerate(tickers):
    file_path = os.path.join(save_dir, f"{ticker}.csv")
    fundamentals_path = os.path.join(FUNDAMENTAL_DIR,
    ↪f"{ticker}_fundamentals.csv")
    try:
        stock = yf.Ticker(ticker)
        if os.path.exists(file_path) and not force_redownload:
            print(f"Reading cached data for {ticker}")
            df = pd.read_csv(file_path)
            if 'Date' not in df.columns or 'Close' not in df.columns:
                continue
            df['Date'] = pd.to_datetime(df['Date'])
            df.set_index('Date', inplace=True)
        else:
            print(f"Downloading {ticker} ({i+1}/{len(tickers)})")
            df = stock.history(start=start, end=end, auto_adjust=False)
            if df.empty or len(df) < 200:
                continue
            df.reset_index(inplace=True)
            df.to_csv(file_path, index=False)
            df.set_index('Date', inplace=True)
        info = stock.info
        new_row = {
            'Date': datetime.today().strftime("%Y-%m-%d"),
            'Ticker': ticker,
            'peRatio': info.get('trailingPE', 0.0),
            'pegRatio': info.get('pegRatio', 0.0),
            'priceToSales': info.get('priceToSalesTrailing12Months', 0.0),
            'priceToBook': info.get('priceToBook', 0.0),
            'debtToEquity': info.get('debtToEquity', 0.0),
            'ebitda': info.get('ebitda', 0.0)
        }
        if os.path.exists(fundamentals_path):
            existing = pd.read_csv(fundamentals_path)
            if not ((existing['Date'] == new_row['Date']) &
            ↪(existing['Ticker'] == new_row['Ticker'])).any():
                updated = pd.concat([existing, pd.DataFrame([new_row])],
            ↪ignore_index=True)
                updated.to_csv(fundamentals_path, index=False)
            else:
                pd.DataFrame([new_row]).to_csv(fundamentals_path, index=False)
            df['Ticker'] = ticker

```

```

        data[ticker] = df
        time.sleep(0.2)
    except Exception as e:
        print(f"Error with {ticker}: {e}")
    return data

# Preprocess for training
def preprocess_data(data):
    frames = []
    for ticker, df in data.items():
        try:
            if 'Close' not in df.columns:
                continue
            df = df[['Open', 'High', 'Low', 'Volume', 'Close']].copy()
            df['Lag_Close'] = df['Close'].shift(1)
            df['MA_5'] = df['Close'].rolling(5).mean()
            df['Return'] = df['Close'].pct_change()
            df['Target'] = df['Close'].shift(-5)
            df.dropna(inplace=True)
            if len(df) < 10:
                continue
            fund_file = os.path.join(FUNDAMENTAL_DIR, f"{ticker}_fundamentals.
↪csv")
            if not os.path.exists(fund_file):
                continue
            row = pd.read_csv(fund_file).sort_values('Date').iloc[-1]
            for col in ['peRatio', 'pegRatio', 'priceToSales', 'priceToBook',
↪'debtToEquity', 'ebitda']:
                value = row.get(col, 0.0)
                df[col] = value if not pd.isna(value) else 0.0
            df['Ticker'] = ticker
            frames.append(df)
        except Exception as e:
            print(f"Preprocessing error on {ticker}: {e}")
    if not frames:
        raise ValueError("All tickers failed during preprocessing.")
    result = pd.concat(frames)
    result.reset_index(inplace=True)
    return result.dropna()

# Extract features and labels
def prepare_features(df):
    y = df['Target']
    X = df.drop(columns=['Target'])
    non_numeric_cols = X.select_dtypes(include=['object', 'datetime64']).
↪columns.tolist()
    for col in ['Date', 'Datetime']:

```

```

        if col in X.columns:
            non_numeric_cols.append(col)
X.drop(columns=list(set(non_numeric_cols)), inplace=True)
le = LabelEncoder()
if 'Ticker' in df.columns:
    X['Ticker'] = le.fit_transform(df['Ticker'])
if y.nunique() <= 1:
    raise ValueError("Target values have no variation.")
return X, y

# Train and evaluate ANN
def build_ann(input_dim):
    model = Sequential([
        InputLayer(input_shape=(input_dim,)),
        Dense(128, activation='relu'),
        Dense(64, activation='relu'),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mse')
    return model

# Train and evaluate CNN
def build_cnn(input_shape):
    model = Sequential([
        InputLayer(input_shape=input_shape),
        Conv1D(32, kernel_size=3, activation='relu'),
        MaxPooling1D(pool_size=2),
        Flatten(),
        Dense(64, activation='relu'),
        Dense(1)
    ])
    model.compile(optimizer='adam', loss='mse')
    return model

def train_ann_cnn(X, y):
    scaler = StandardScaler()
    X_scaled = scaler.fit_transform(X)
    X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
        shuffle=False, test_size=0.2)

    ann = build_ann(X_train.shape[1])
    ann.fit(X_train, y_train, epochs=30, batch_size=64, verbose=0)
    ann_preds = ann.predict(X_test).flatten()
    y_test_np = np.array(y_test)
    ann_mask = ~np.isnan(ann_preds) & ~np.isnan(y_test_np)
    ann_preds = ann_preds[ann_mask]
    ann_clean = y_test_np[ann_mask]

```

```

ann_rmse = mean_squared_error(ann_clean, ann_preds, squared=False)
ann_r2 = r2_score(ann_clean, ann_preds)
print("ANN pred stats:", np.min(ann_preds), np.max(ann_preds), np.
↳mean(ann_preds))

X_cnn_train = X_train.reshape((X_train.shape[0], X_train.shape[1], 1))
X_cnn_test = X_test.reshape((X_test.shape[0], X_test.shape[1], 1))
cnn = build_cnn((X_cnn_train.shape[1], 1))
cnn.fit(X_cnn_train, y_train, epochs=30, batch_size=64, verbose=0)
cnn_preds = cnn.predict(X_cnn_test).flatten()
mask = ~np.isnan(cnn_preds) & ~np.isnan(y_test_np)
cnn_preds = cnn_preds[mask]
y_clean = y_test_np[mask]
if len(y_clean) == 0:
    cnn_rmse = np.nan
    cnn_r2 = np.nan
    print("CNN predictions invalid: all values filtered out.")
else:
    cnn_rmse = mean_squared_error(y_clean, cnn_preds, squared=False)
    cnn_r2 = r2_score(y_clean, cnn_preds)
    print("CNN pred stats:", np.min(cnn_preds), np.max(cnn_preds), np.
↳mean(cnn_preds))

    return ann_clean, ann_preds, y_clean, cnn_preds, ann_rmse, ann_r2,
↳cnn_rmse, cnn_r2

# Plot
def plot_predictions(y_true_ann, ann_preds, y_true_cnn, cnn_preds):
    if len(y_true_ann) == len(ann_preds):
        plt.figure(figsize=(14, 6))
        plt.plot(y_true_ann, label='Actual', linewidth=2)
        plt.plot(ann_preds, label='ANN Prediction', alpha=0.7)
        plt.title("ANN Predictions vs Actual")
        plt.xlabel("Index")
        plt.ylabel("Price")
        plt.legend()
        plt.grid(True)
        plt.tight_layout()
        plt.show()
    else:
        print("Skipping ANN plot: mismatched lengths.")

    if len(y_true_cnn) == len(cnn_preds) and len(y_true_cnn) > 0:
        plt.figure(figsize=(14, 6))
        plt.plot(y_true_cnn, label='Actual', linewidth=2)
        plt.plot(cnn_preds, label='CNN Prediction', alpha=0.7)
        plt.title("CNN Predictions vs Actual")

```

```

        plt.xlabel("Index")
        plt.ylabel("Price")
        plt.legend()
        plt.grid(True)
        plt.tight_layout()
        plt.show()
    else:
        print("Skipping CNN plot: predictions invalid or empty.")

def plot_scatter(y_true, y_pred, model_name="Model"):
    plt.figure(figsize=(6, 6))
    plt.scatter(y_true, y_pred, alpha=0.4)
    plt.plot([y_true.min(), y_true.max()], [y_true.min(), y_true.max()], '--',
        ↪color='gray')
    plt.xlabel("Actual Price")
    plt.ylabel("Predicted Price")
    plt.title(f"{model_name}: Predicted vs Actual")
    plt.grid(True)
    plt.tight_layout()
    plt.show()

# Main
tickers = get_sp500_tickers()[ :NUM_COMPANIES]
stock_data = download_stock_data(tickers, force_redownload=FORCE_REDOWNLOAD)
processed = preprocess_data(stock_data)
X, y = prepare_features(processed)
ann_clean, ann_pred, y_clean, cnn_pred, ann_rmse, ann_r2, cnn_rmse, cnn_r2 = ↪
    ↪train_ann_cnn(X, y)

plot_predictions(ann_clean, ann_pred, y_clean, cnn_pred)
plot_scatter(ann_clean, ann_pred, model_name="ANN")
plot_scatter(y_clean, cnn_pred, model_name="CNN")

print(f"\nANN: RMSE = {ann_rmse:.2f}")
print(f"ANN: R2 Score = {ann_r2:.4f}")
print(f"ANN: MAE = {mean_absolute_error(ann_clean, ann_pred):.2f}")
print(f"\nCNN: RMSE = {cnn_rmse:.2f}")
print(f"CNN: R2 Score = {cnn_r2:.4f}")
print(f"CNN: MAE = {mean_absolute_error(y_clean, cnn_pred):.2f}")

```

```

Downloading MMM (1/500)
Downloading AOS (2/500)
Downloading ABT (3/500)
Downloading ABBV (4/500)
Downloading ACN (5/500)
Downloading ADBE (6/500)
Downloading AMD (7/500)
Downloading AES (8/500)

```

Downloading AFL (9/500)
Downloading A (10/500)
Downloading APD (11/500)
Downloading ABNB (12/500)
Downloading AKAM (13/500)
Downloading ALB (14/500)
Downloading ARE (15/500)
Downloading ALGN (16/500)
Downloading ALLE (17/500)
Downloading LNT (18/500)
Downloading ALL (19/500)
Downloading GOOGL (20/500)
Downloading GOOG (21/500)
Downloading MO (22/500)
Downloading AMZN (23/500)
Downloading AMCR (24/500)
Downloading AEE (25/500)
Downloading AEP (26/500)
Downloading AXP (27/500)
Downloading AIG (28/500)
Downloading AMT (29/500)
Downloading AWK (30/500)
Downloading AMP (31/500)
Downloading AME (32/500)
Downloading AMGN (33/500)
Downloading APH (34/500)
Downloading ADI (35/500)
Downloading AON (36/500)
Downloading APA (37/500)
Downloading APO (38/500)
Downloading AAPL (39/500)
Downloading AMAT (40/500)
Downloading APTV (41/500)
Downloading ACGL (42/500)
Downloading ADM (43/500)
Downloading ANET (44/500)
Downloading AJG (45/500)
Downloading AIZ (46/500)
Downloading T (47/500)
Downloading ATQ (48/500)
Downloading ADSK (49/500)
Downloading ADP (50/500)
Downloading AZO (51/500)
Downloading AVB (52/500)
Downloading AVY (53/500)
Downloading AXON (54/500)
Downloading BKR (55/500)
Downloading BALL (56/500)

Downloading BAC (57/500)
Downloading BAX (58/500)
Downloading BDX (59/500)
Downloading BRK-B (60/500)
Downloading BBY (61/500)
Downloading TECH (62/500)
Downloading BIIB (63/500)
Downloading BLK (64/500)
Downloading BX (65/500)
Downloading XYZ (66/500)
Downloading BK (67/500)
Downloading BA (68/500)
Downloading BKNG (69/500)
Downloading BSX (70/500)
Downloading BMY (71/500)
Downloading AVGO (72/500)
Downloading BR (73/500)
Downloading BRO (74/500)
Downloading BF-B (75/500)
Downloading BLDR (76/500)
Downloading BG (77/500)
Downloading BXP (78/500)
Downloading CHRW (79/500)
Downloading CDNS (80/500)
Downloading CZR (81/500)
Downloading CPT (82/500)
Downloading CPB (83/500)
Downloading COF (84/500)
Downloading CAH (85/500)
Downloading KMX (86/500)
Downloading CCL (87/500)
Downloading CARR (88/500)
Downloading CAT (89/500)
Downloading CBOE (90/500)
Downloading CBRE (91/500)
Downloading CDW (92/500)
Downloading COR (93/500)
Downloading CNC (94/500)
Downloading CNP (95/500)
Downloading CF (96/500)
Downloading CRL (97/500)
Downloading SCHW (98/500)
Downloading CHTR (99/500)
Downloading CVX (100/500)
Downloading CMG (101/500)
Downloading CB (102/500)
Downloading CHD (103/500)
Downloading CI (104/500)

Downloading CINF (105/500)
Downloading CTAS (106/500)
Downloading CSCD (107/500)
Downloading C (108/500)
Downloading CFG (109/500)
Downloading CLX (110/500)
Downloading CME (111/500)
Downloading CMS (112/500)
Downloading KO (113/500)
Downloading CTSH (114/500)
Downloading COIN (115/500)
Downloading CL (116/500)
Downloading CMCSA (117/500)
Downloading CAG (118/500)
Downloading COP (119/500)
Downloading ED (120/500)
Downloading STZ (121/500)
Downloading CEG (122/500)
Downloading COO (123/500)
Downloading CPRT (124/500)
Downloading GLW (125/500)
Downloading CPAY (126/500)
Downloading CTVA (127/500)
Downloading CSGP (128/500)
Downloading COST (129/500)
Downloading CTRA (130/500)
Downloading CRWD (131/500)
Downloading CCI (132/500)
Downloading CSX (133/500)
Downloading CMI (134/500)
Downloading CVS (135/500)
Downloading DHR (136/500)
Downloading DRI (137/500)
Downloading DDOG (138/500)
Downloading DVA (139/500)
Downloading DAY (140/500)
Downloading DECK (141/500)
Downloading DE (142/500)
Downloading DELL (143/500)
Downloading DAL (144/500)
Downloading DVN (145/500)
Downloading DXCM (146/500)
Downloading FANG (147/500)
Downloading DLR (148/500)
Downloading DG (149/500)
Downloading DLTR (150/500)
Downloading D (151/500)
Downloading DPZ (152/500)

Downloading DASH (153/500)
Downloading DOV (154/500)
Downloading DOW (155/500)
Downloading DHI (156/500)
Downloading DTE (157/500)
Downloading DUK (158/500)
Downloading DD (159/500)
Downloading EMN (160/500)
Downloading ETN (161/500)
Downloading EBAY (162/500)
Downloading ECL (163/500)
Downloading EIX (164/500)
Downloading EW (165/500)
Downloading EA (166/500)
Downloading ELV (167/500)
Downloading EMR (168/500)
Downloading ENPH (169/500)
Downloading ETR (170/500)
Downloading EOG (171/500)
Downloading EPAM (172/500)
Downloading EQT (173/500)
Downloading EFX (174/500)
Downloading EQIX (175/500)
Downloading EQR (176/500)
Downloading ERIE (177/500)
Downloading ESS (178/500)
Downloading EL (179/500)
Downloading EG (180/500)
Downloading EVRG (181/500)
Downloading ES (182/500)
Downloading EXC (183/500)
Downloading EXE (184/500)
Downloading EXPE (185/500)
Downloading EXPD (186/500)
Downloading EXR (187/500)
Downloading XOM (188/500)
Downloading FFIV (189/500)
Downloading FDS (190/500)
Downloading FICO (191/500)
Downloading FAST (192/500)
Downloading FRT (193/500)
Downloading FDX (194/500)
Downloading FIS (195/500)
Downloading FITB (196/500)
Downloading FSLR (197/500)
Downloading FE (198/500)
Downloading FI (199/500)
Downloading F (200/500)

Downloading FTNT (201/500)
Downloading FTV (202/500)
Downloading FOXA (203/500)
Downloading FOX (204/500)
Downloading BEN (205/500)
Downloading FCX (206/500)
Downloading GRMN (207/500)
Downloading IT (208/500)
Downloading GE (209/500)
Downloading GEHC (210/500)
Downloading GEV (211/500)
Downloading GEN (212/500)
Downloading GNRC (213/500)
Downloading GD (214/500)
Downloading GIS (215/500)
Downloading GM (216/500)
Downloading GPC (217/500)
Downloading GILD (218/500)
Downloading GPN (219/500)
Downloading GL (220/500)
Downloading GDDY (221/500)
Downloading GS (222/500)
Downloading HAL (223/500)
Downloading HIG (224/500)
Downloading HAS (225/500)
Downloading HCA (226/500)
Downloading DOC (227/500)
Downloading HSIC (228/500)
Downloading HSY (229/500)
Downloading HPE (230/500)
Downloading HLT (231/500)
Downloading HOLX (232/500)
Downloading HD (233/500)
Downloading HON (234/500)
Downloading HRL (235/500)
Downloading HST (236/500)
Downloading HWM (237/500)
Downloading HPQ (238/500)
Downloading HUBB (239/500)
Downloading HUM (240/500)
Downloading HBAN (241/500)
Downloading HII (242/500)
Downloading IBM (243/500)
Downloading IEX (244/500)
Downloading IDXX (245/500)
Downloading ITW (246/500)
Downloading INCY (247/500)
Downloading IR (248/500)

Downloading PDD (249/500)
Downloading INTC (250/500)
Downloading ICE (251/500)
Downloading IFF (252/500)
Downloading IP (253/500)
Downloading IPG (254/500)
Downloading INTU (255/500)
Downloading ISRG (256/500)
Downloading IVZ (257/500)
Downloading INVH (258/500)
Downloading IQV (259/500)
Downloading IRM (260/500)
Downloading JBHT (261/500)
Downloading JBL (262/500)
Downloading JKHY (263/500)
Downloading J (264/500)
Downloading JNJ (265/500)
Downloading JCI (266/500)
Downloading JPM (267/500)
Downloading K (268/500)
Downloading KVUE (269/500)
Downloading KDP (270/500)
Downloading KEY (271/500)
Downloading KEYS (272/500)
Downloading KMB (273/500)
Downloading KIM (274/500)
Downloading KMI (275/500)
Downloading KKR (276/500)
Downloading KLAC (277/500)
Downloading KHC (278/500)
Downloading KR (279/500)
Downloading LHX (280/500)
Downloading LH (281/500)
Downloading LRCX (282/500)
Downloading LW (283/500)
Downloading LVS (284/500)
Downloading LDOS (285/500)
Downloading LEN (286/500)
Downloading LII (287/500)
Downloading LLY (288/500)
Downloading LIN (289/500)
Downloading LYV (290/500)
Downloading LKQ (291/500)
Downloading LMT (292/500)
Downloading L (293/500)
Downloading LOW (294/500)
Downloading LULU (295/500)
Downloading LYB (296/500)

Downloading MTB (297/500)
Downloading MPC (298/500)
Downloading MKTX (299/500)
Downloading MAR (300/500)
Downloading MMC (301/500)
Downloading MLM (302/500)
Downloading MAS (303/500)
Downloading MA (304/500)
Downloading MTCH (305/500)
Downloading MKC (306/500)
Downloading MCD (307/500)
Downloading MCK (308/500)
Downloading MDT (309/500)
Downloading MRK (310/500)
Downloading META (311/500)
Downloading MET (312/500)
Downloading MTD (313/500)
Downloading MGM (314/500)
Downloading MCHP (315/500)
Downloading MU (316/500)
Downloading MSFT (317/500)
Downloading MAA (318/500)
Downloading MRNA (319/500)
Downloading MHK (320/500)
Downloading MOH (321/500)
Downloading TAP (322/500)
Downloading MDLZ (323/500)
Downloading MPWR (324/500)
Downloading MNST (325/500)
Downloading MCO (326/500)
Downloading MS (327/500)
Downloading MOS (328/500)
Downloading MSI (329/500)
Downloading MSCI (330/500)
Downloading NDAQ (331/500)
Downloading NTAP (332/500)
Downloading NFLX (333/500)
Downloading NEM (334/500)
Downloading NWSA (335/500)
Downloading NWS (336/500)
Downloading NEE (337/500)
Downloading NKE (338/500)
Downloading NI (339/500)
Downloading NDSN (340/500)
Downloading NSC (341/500)
Downloading NTRS (342/500)
Downloading NOC (343/500)
Downloading NCLH (344/500)

Downloading NRG (345/500)
Downloading NUE (346/500)
Downloading NVDA (347/500)
Downloading NVR (348/500)
Downloading NXPI (349/500)
Downloading ORLY (350/500)
Downloading OXY (351/500)
Downloading ODFL (352/500)
Downloading OMC (353/500)
Downloading ON (354/500)
Downloading OKE (355/500)
Downloading ORCL (356/500)
Downloading OTIS (357/500)
Downloading PCAR (358/500)
Downloading PKG (359/500)
Downloading PLTR (360/500)
Downloading PANW (361/500)
Downloading PARA (362/500)
Downloading PH (363/500)
Downloading PAYX (364/500)
Downloading PAYC (365/500)
Downloading PYPL (366/500)
Downloading PNR (367/500)
Downloading PEP (368/500)
Downloading PFE (369/500)
Downloading PCG (370/500)
Downloading PM (371/500)
Downloading PSX (372/500)
Downloading PNW (373/500)
Downloading PNC (374/500)
Downloading POOL (375/500)
Downloading PPG (376/500)
Downloading PPL (377/500)
Downloading PFG (378/500)
Downloading PG (379/500)
Downloading PGR (380/500)
Downloading PLD (381/500)
Downloading PRU (382/500)
Downloading PEG (383/500)
Downloading PTC (384/500)
Downloading PSA (385/500)
Downloading PHM (386/500)
Downloading PWR (387/500)
Downloading QCOM (388/500)
Downloading DGX (389/500)
Downloading RL (390/500)
Downloading RJF (391/500)
Downloading RTX (392/500)

Downloading O (393/500)
Downloading REG (394/500)
Downloading REGN (395/500)
Downloading RF (396/500)
Downloading RSG (397/500)
Downloading RMD (398/500)
Downloading RVTY (399/500)
Downloading ROK (400/500)
Downloading ROL (401/500)
Downloading ROP (402/500)
Downloading ROST (403/500)
Downloading RCL (404/500)
Downloading SPGI (405/500)
Downloading CRM (406/500)
Downloading SBAC (407/500)
Downloading SLB (408/500)
Downloading STX (409/500)
Downloading SRE (410/500)
Downloading NOW (411/500)
Downloading SHW (412/500)
Downloading SPG (413/500)
Downloading SWKS (414/500)
Downloading SJM (415/500)
Downloading SW (416/500)
Downloading SNA (417/500)
Downloading SOLV (418/500)
Downloading SO (419/500)
Downloading LUV (420/500)
Downloading SWK (421/500)
Downloading SBUX (422/500)
Downloading STT (423/500)
Downloading STLD (424/500)
Downloading STE (425/500)
Downloading SYK (426/500)
Downloading SMCI (427/500)
Downloading SYF (428/500)
Downloading SNPS (429/500)
Downloading SYY (430/500)
Downloading TMUS (431/500)
Downloading TROW (432/500)
Downloading TTWO (433/500)
Downloading TPR (434/500)
Downloading TRGP (435/500)
Downloading TGT (436/500)
Downloading TEL (437/500)
Downloading TDY (438/500)
Downloading TER (439/500)
Downloading TSLA (440/500)

Downloading TXN (441/500)
Downloading TPL (442/500)
Downloading TXT (443/500)
Downloading TMO (444/500)
Downloading TJX (445/500)
Downloading TKO (446/500)
Downloading TTD (447/500)
Downloading TSCO (448/500)
Downloading TT (449/500)
Downloading TDG (450/500)
Downloading TRV (451/500)
Downloading TRMB (452/500)
Downloading TFC (453/500)
Downloading TYL (454/500)
Downloading TSN (455/500)
Downloading USB (456/500)
Downloading UBER (457/500)
Downloading UDR (458/500)
Downloading ULTA (459/500)
Downloading UNP (460/500)
Downloading UAL (461/500)
Downloading UPS (462/500)
Downloading URI (463/500)
Downloading UNH (464/500)
Downloading UHS (465/500)
Downloading VLO (466/500)
Downloading VTR (467/500)
Downloading VLTO (468/500)
Downloading VRSN (469/500)
Downloading VRSK (470/500)
Downloading VZ (471/500)
Downloading VRTX (472/500)
Downloading VTRS (473/500)
Downloading VICI (474/500)
Downloading V (475/500)
Downloading VST (476/500)
Downloading VMC (477/500)
Downloading WRB (478/500)
Downloading GWW (479/500)
Downloading WAB (480/500)
Downloading WBA (481/500)
Downloading WMT (482/500)
Downloading DIS (483/500)
Downloading WBD (484/500)
Downloading WM (485/500)
Downloading WAT (486/500)
Downloading WEC (487/500)
Downloading WFC (488/500)

Downloading WELL (489/500)
Downloading WST (490/500)
Downloading WDC (491/500)
Downloading WY (492/500)
Downloading WSM (493/500)
Downloading WMB (494/500)
Downloading WTW (495/500)
Downloading WDAY (496/500)
Downloading WYNN (497/500)
Downloading XEL (498/500)
Downloading XYL (499/500)
Downloading YUM (500/500)

```
C:\Users\Steve\anaconda3\Lib\site-  
packages\keras\src\layers\core\input_layer.py:27: UserWarning: Argument  
`input_shape` is deprecated. Use `shape` instead.  
    warnings.warn(
```

17682/17682 5s 283us/step

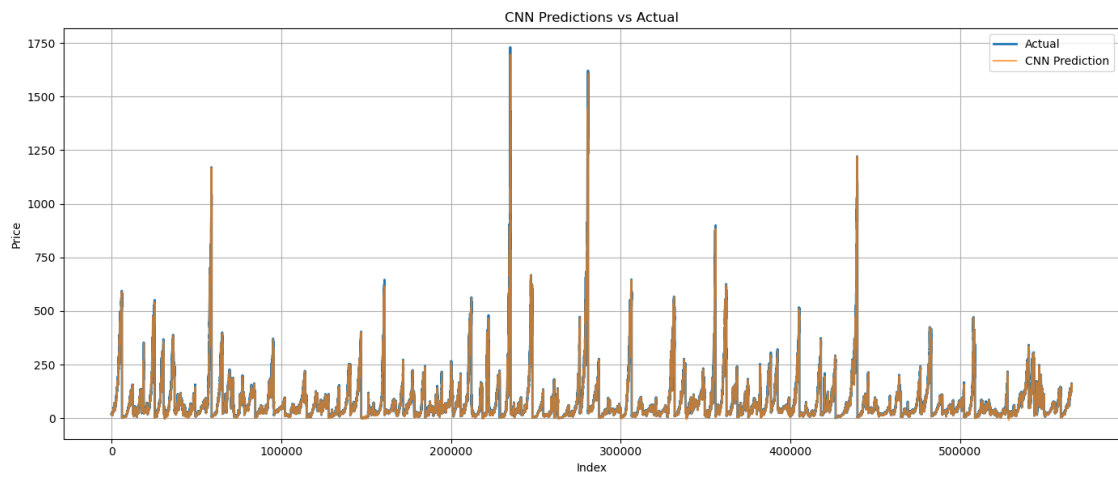
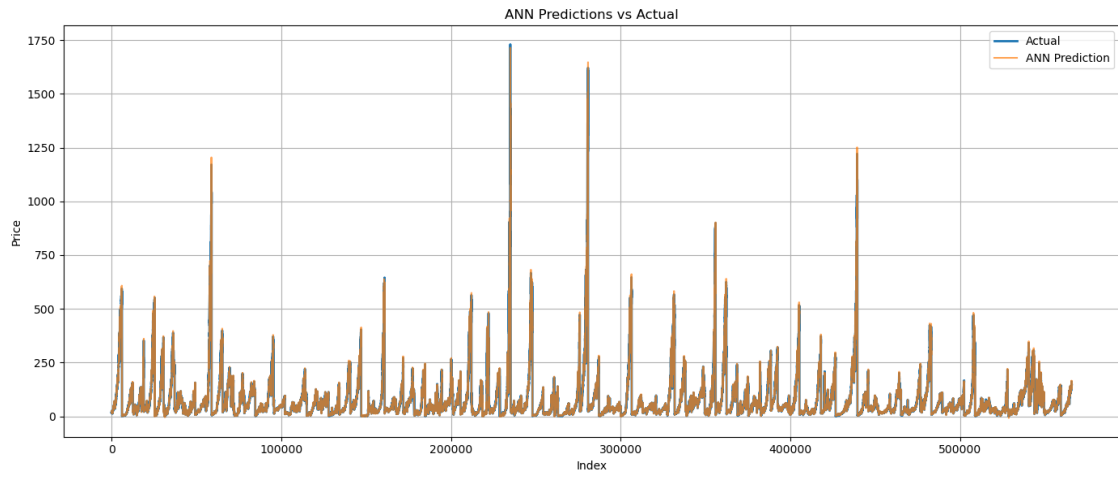
```
C:\Users\Steve\anaconda3\Lib\site-packages\sklearn\metrics\_regression.py:492:  
FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in  
1.6. To calculate the root mean squared error, use the  
function'root_mean_squared_error'.  
    warnings.warn(
```

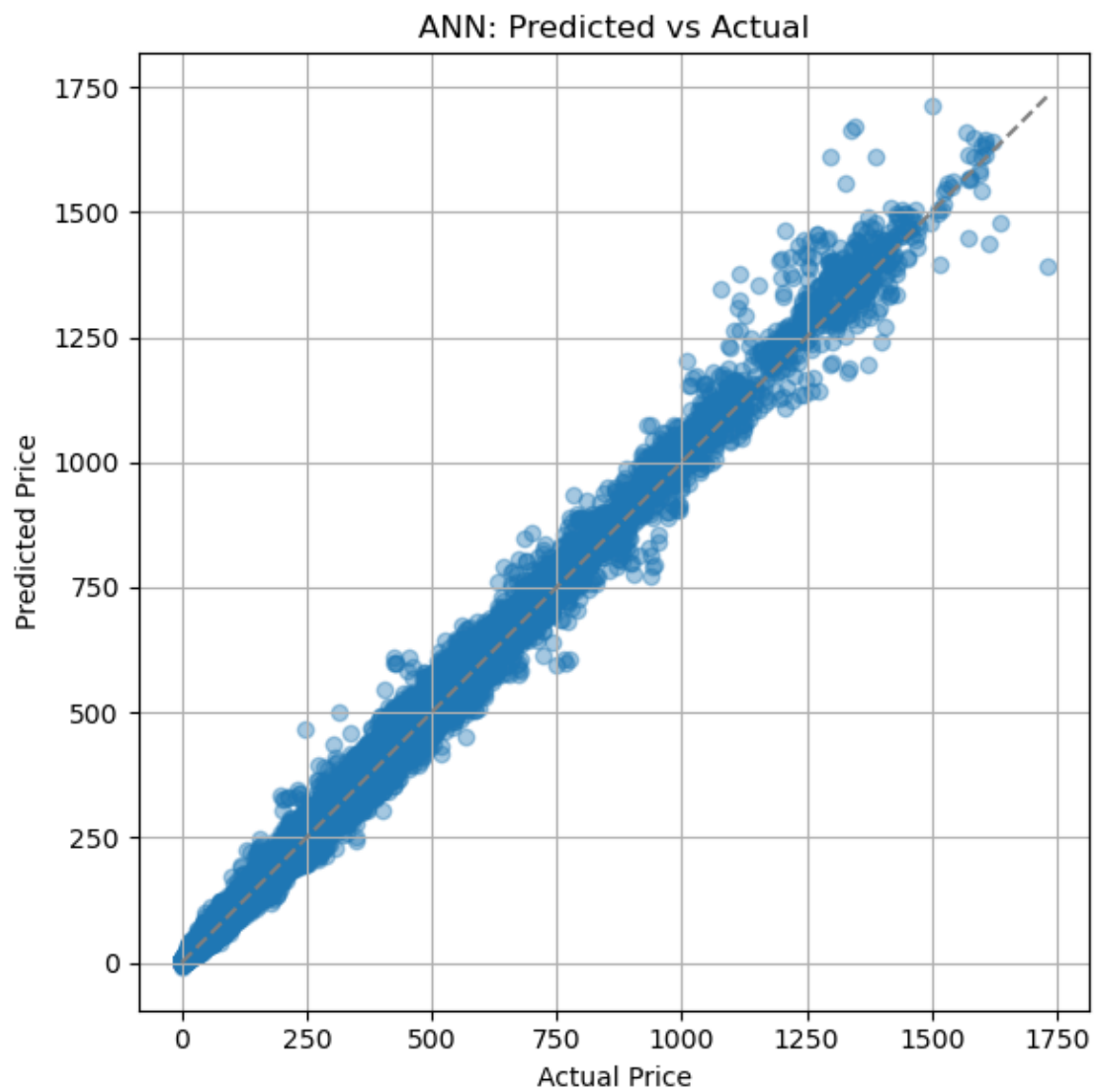
```
C:\Users\Steve\anaconda3\Lib\site-  
packages\keras\src\layers\core\input_layer.py:27: UserWarning: Argument  
`input_shape` is deprecated. Use `shape` instead.  
    warnings.warn(
```

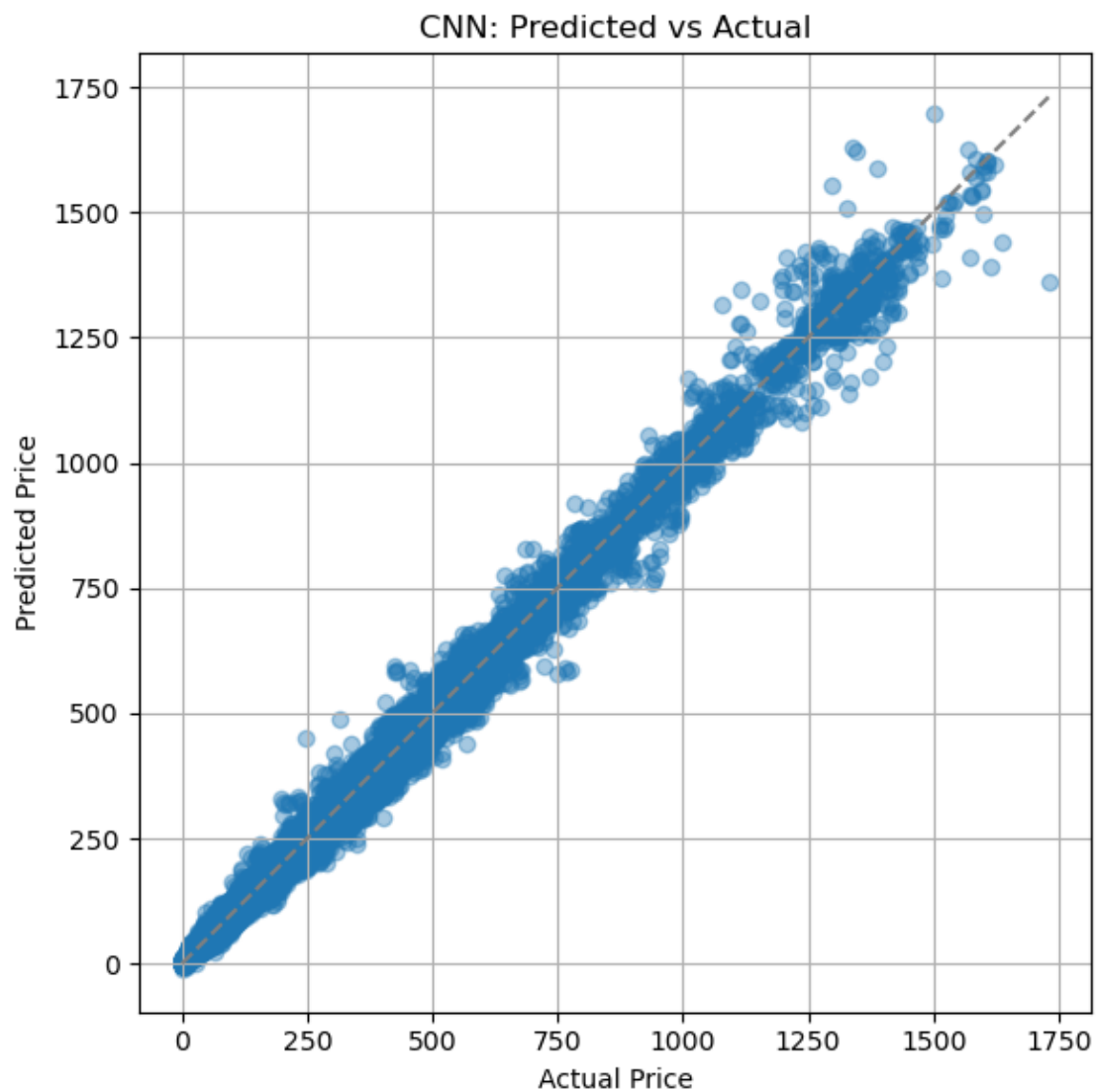
ANN pred stats: -7.7171116 1713.7443 79.07099
17682/17682 7s 401us/step

```
C:\Users\Steve\anaconda3\Lib\site-packages\sklearn\metrics\_regression.py:492:  
FutureWarning: 'squared' is deprecated in version 1.4 and will be removed in  
1.6. To calculate the root mean squared error, use the  
function'root_mean_squared_error'.  
    warnings.warn(
```

CNN pred stats: -9.51735 1695.5841 76.1013







ANN: RMSE = 6.36

ANN: R2 Score = 0.9965

ANN: MAE = 2.76

CNN: RMSE = 6.19

CNN: R2 Score = 0.9967

CNN: MAE = 3.10