Analysis of High-Frequency Data with Wavelet Transforms

Willie Bedoya

Problem Formulation and Background

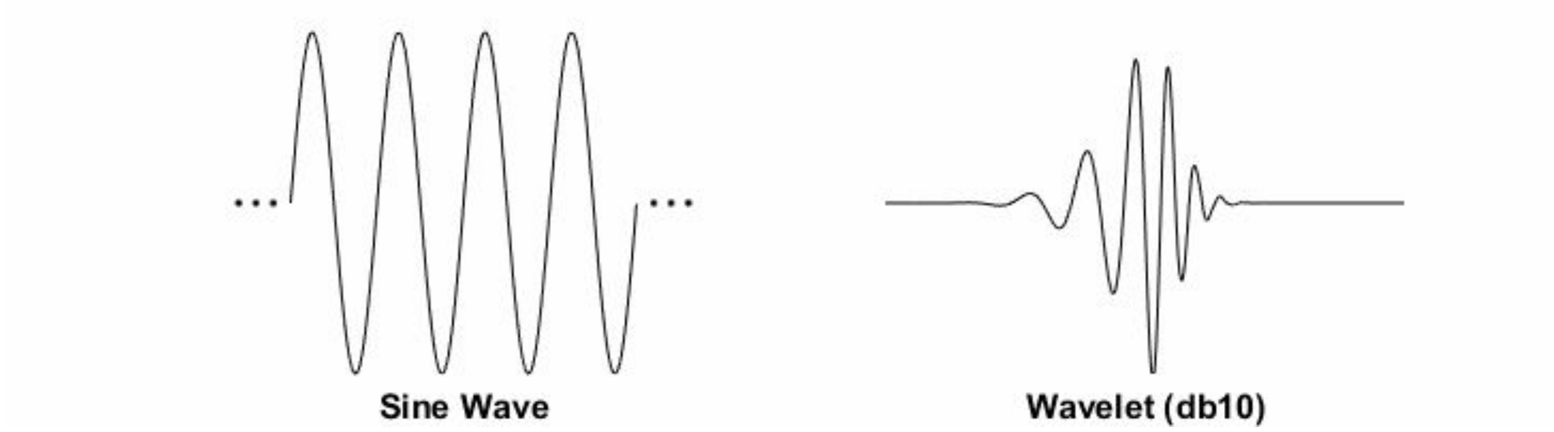
- The objective of this project is to assess the quality of denoised price signals for financial time series.
- Methodology:
  - Collect high-frequency hourly price data from five stocks (TSLA, WMT, JNJ, INTC, AAPL) between January 5, 2016 and April 11, 2024, totaling over 14,000 data points per stock.
  - Add Gaussian noise to the collected data to mimic real-world market fluctuations.
  - Apply wavelet transforms to clean the noise-added price data.
- Analysis:
  - Evaluate the quality of the denoised signals against the original noisy data.
  - Determine the effectiveness of different wavelet families and thresholding methods.
- Why not use Fourier transforms?
  - The reason is related to Heisenberg's uncertainty principle: Fourier transforms are localized in frequency, and there is some trade-off between measuring very small differences in both frequency and time
  - Wavelet denoising seeks to take advantage of the in-between case, as it is localized in not only frequency but also time

Problem Formulation and Background

- Definition of a wavelet:
  - A waveform with limited duration, an average value of zero, and a nonzero norm.
  - Image on next slide shows limited duration and irregular shape of wavelets versus sine waves.
- General form of continuous wavelet transform: $\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\Psi(\frac{t-b}{a}),\ a, b, t \in (-\infty, \infty)$
  - Where *a* is a scaling factor (adjusts width of wavelet) and *b* is a time shift factor (moves wavelet along time axis)
  - Two conditions must be satisfied: zero mean and admissibility (ensures wavelet can be used to construct original signal)

$$\int_{-\infty}^{\infty} \Psi(t)dt = 0 \qquad C_{\Psi} = \int_{-\infty}^{\infty} \frac{|\widehat{\Psi}(\omega)|^2}{|\omega|} d\omega < \infty$$

Comparison: Sine Wave vs. Wavelet



Approach and Description of Solution Methods

- Numerous methods for wavelet denoising
  - Improved Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (ICEEMDAN)
  - Discrete wavelet transforms
- Let's start with ICEEMDAN:
  - Frst, Gaussian noise is added to the original time series $x(t)$ to form $x_i(t)$
  - Where β_0 is an amplitude coefficient, $E_1(\cdot)$ is the first intrinsic mode function (IMF) component obtained, and ω_i is the i-th added Gaussian white noise.
  - Then, obtain the first stage residual $r_1(t)$ by calculating the local average $M(x_i(t))$ of the noise-added time series. Common approach used here for the local average operator $M(\cdot)$ is cubic spline interpolation.

$$x_i(t) = x(t) + \beta_0 E_1(\omega^i) \qquad r_1(t) = < M(x_i(t)) >$$

Approach and Description of Solution Methods

- The first IMF of the time series is found as $IMF_1(t)$

$$IMF_1(t) = x(t) - r_1(t)$$

- Now, using this first stage residual the local average of a new signal $r_1(t) + \beta_1 E_2(\omega_i)$ can be found

$$r_2(t) = \; < M(r_1(t) + \beta_1 E_2(\omega^i)) >$$

- And the second IMF can be taken…

$$IMF_2(t) = r_1(t) - r_2(t)$$

- Notice that this is a recursive formula, which can be written as follows:

$$r_k(t) = \; < M(r_{k-1}(t) + \beta_{k-1} E_k(\omega^i)) > \qquad IMF_k(t) = r_{k-1}(t) - r_k(t)$$

Approach and Description of Solution Methods

- These iterations should be repeated until the residual does not pass a certain threshold. Then, the residual is equal to

$$R(t) = x(t) - \sum_{k=1}^{K} IMF_k$$

- Rearranging, can find the original time series in terms of IMFs and residual:

$$x(t) = \sum_{k=1}^{K} IMF_k + R(t)$$

- For every IMF component and the sum of IMFs components, two metrics should be ensured
  - *The* mean of IMFs is equal to 0
  - Augmented Dicker-Fulley tests to ensure the IMFs are stationary

Approach and Description of Solution Methods

- The IMFs and residual are now separated into two different components: $x(t)_{noise}$ and $x(t)_{non\text{-}noise}$

$$x(t)_{noise} = \sum_{k=1}^{i} IMF_k \qquad\qquad x(t)_{non-noise} = \sum_{k=i+1}^{K} IMF_k + R(t)$$

- The IMFs which were found to have means of zero and stationarity are summed in the $x(t)_{noise}$ component, while the rest are included in the $x(t)_{non\text{-}noise}$ component
- Now, we take the $x(t)_{noise}$ component and apply wavelet threshold denoising
- After denoising is applied, the component $x(t)_{noise}$ can then be broken into a noise component $\varepsilon(t)$ and a noise-free component $x(t)_{noise\text{-}free}$
- Combining the noise-free $x(t)_{noise\text{-}free}$ and earlier non-noise $x(t)_{non\text{-}noise}$ components creates a de-noised component $x(t)_{de\text{-}noised}$ plus the noise component $\varepsilon(t)$.

$$x(t) = \varepsilon(t) + x(t)_{de-noised}$$

Approach and Description of Solution Methods

- Second wavelet denoising method: discrete wavelet transforms
  - Broadly, involves three steps: (1) wavelet decomposition, (2) thresholding, and (3) reconstruction
- Common iterative algorithm used is called orthogonal-pyramid algorithm
- Two coefficients and two filters:
  - *a* the scale index
  - *b* the location index
  - *G* the lowpass filter (a vector of filter coefficients)
  - *H* the highpass filter (also a vector of filter coefficients)
- Start by setting the first scaling coefficients (i.e., when scale index *a* is equal to 0) equal to the original time series, where *N* is the number of points in the time series

$$A_{a,b} = A_{0,b} = x(t_b)$$
$$b = \{1, 2, 3, \dots, N\}$$

Approach and Description of Solution Methods

- The filters *H* and *G* are considered to be quadrature mirror filters, which means that every second element of *G* is equal to the negative *m*-*p*+1-th element of *H*:

$$g_p = (-1)^p h_{m-p+1}$$
$$p = \{1, 2, 3, \dots, m\}$$

  where *m* is the total number of elements for each filter g_p is the *p*-th element of *G*, and h_{m-p+1} is the *m*-*p*+1-th element of *H*

- To obtain the scaling coefficients $A_{a,b}$ beyond the first values from $A_{0,b}$, the lowpass filter *G* is multiplied with previous smaller-scale coefficients (i.e., the scale index *a* is lower at these levels):

$$A_{a,b} = \sum_{p=1}^{m} g_p A_{(a-1),(2b+m-1-p)}$$

  where b = 1 through $2^{-a}N$

Approach and Description of Solution Methods

- Last two steps are to find the wavelet coefficients and reconstruct the original time series
- The wavelet coefficients $D_{a,b}$ are found similarly to how the scaling coefficients were found, except now $A_{a,b}$ is being multiplied by elements of the highpass filter H instead of G:

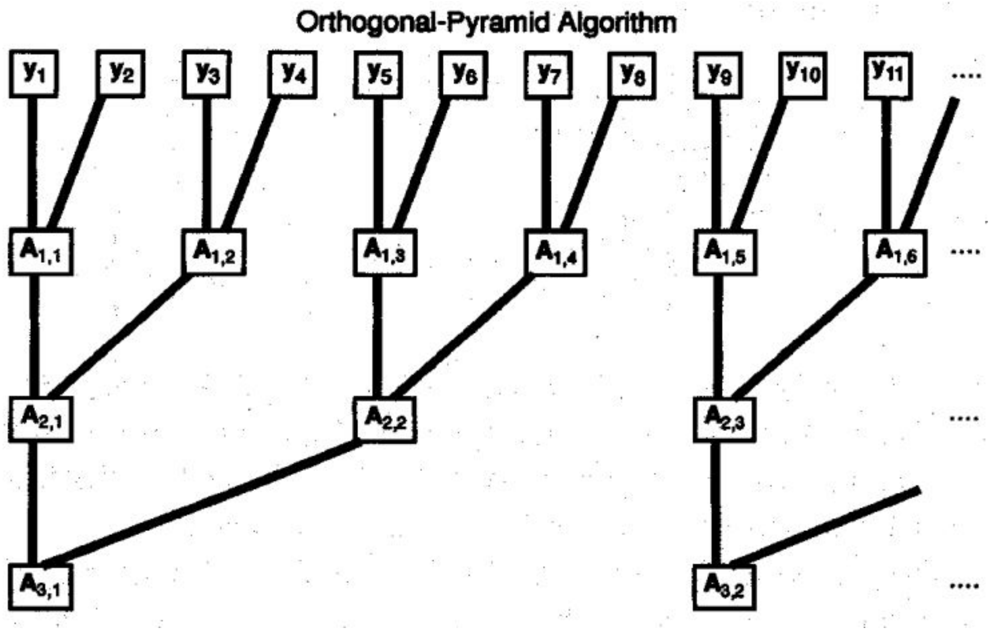
$$D_{a,b} = \sum_{p=1}^{m} h_p A_{(a-1),(2b+m-1-p)}$$

- Once all of the scale indices a and location indices b have been looped through, the time series can be reconstructed by going backwards (i,e., starting from higher scale indices a and going lower):

$$A_{a,b} = \sum_{p=1}^{m} (D_{a+1,b} g_p + A_{a+1,b} h_p)$$

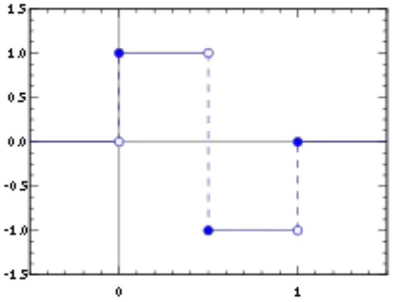
- Once $A_{a,b}$ reaches $A_{0,b}$, recall that this is where we originally started and took as the original time series. Now, it is the discrete wavelet transformed time series.

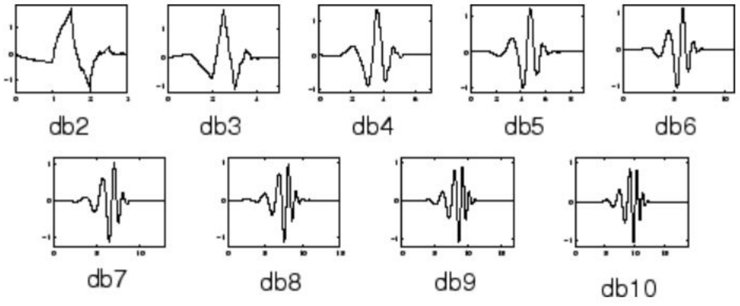
Approach and Description of Solution Methods

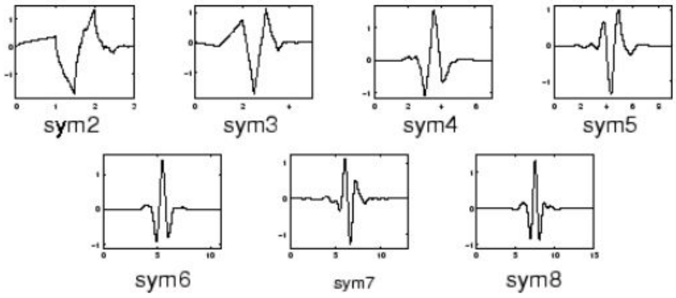


Approach and Description of Solution Methods

- ❖ There are many different wavelet families
  - ➢ Three examples are the Haar, Daubechies, and Symlet wavelets







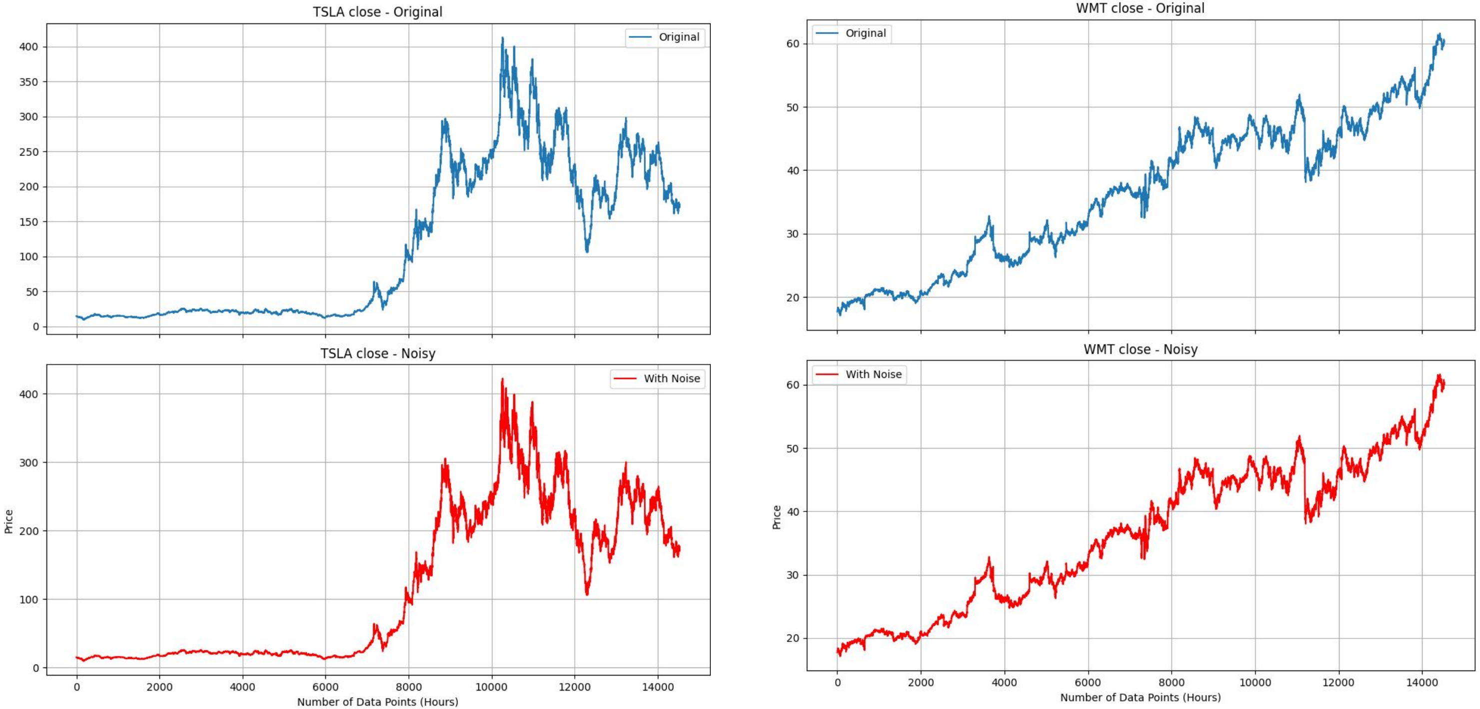
Approach and Description of Solution Methods

- Two main kinds of thresholding (although many variations of these two):
  - Hard thresholding
  - Soft thresholding

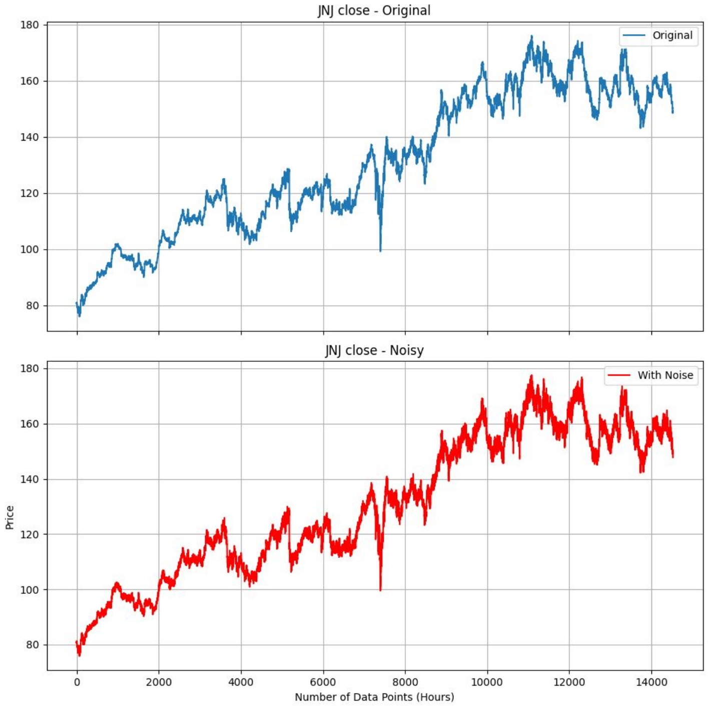
$$\hat{\omega}_{j,k} = \begin{cases} \omega_{j,k}, & \left|\omega_{j,k}\right| \geq \lambda \\ 0, & \left|\omega_{j,k}\right| < \lambda \end{cases}$$

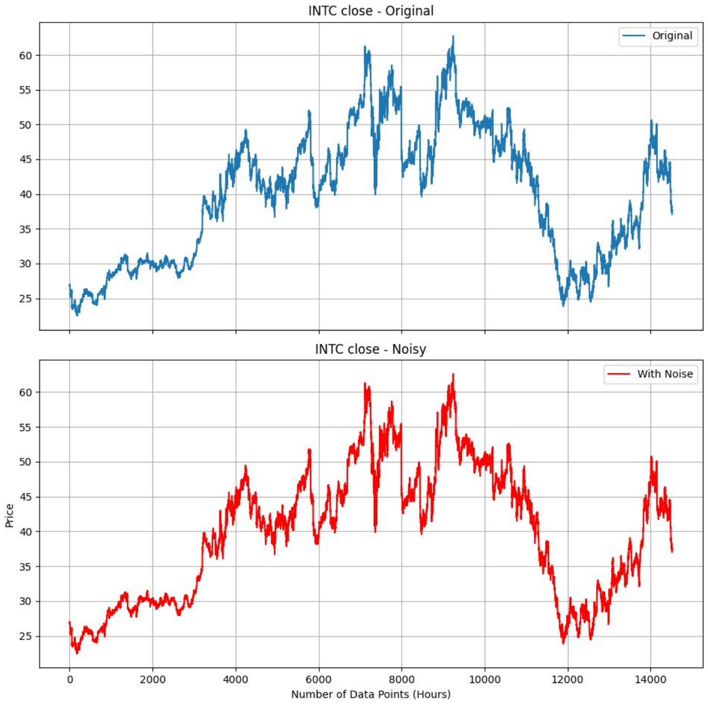
$$\hat{\omega}_{j,k} = \begin{cases} \operatorname{sgn}(\omega_{j,k})\left(\left|\omega_{j,k}\right| - \lambda\right), & \left|\omega_{j,k}\right| \geq \lambda \\ 0, & \left|\omega_{j,k}\right| < \lambda \end{cases}$$

Results (Adding Noise)

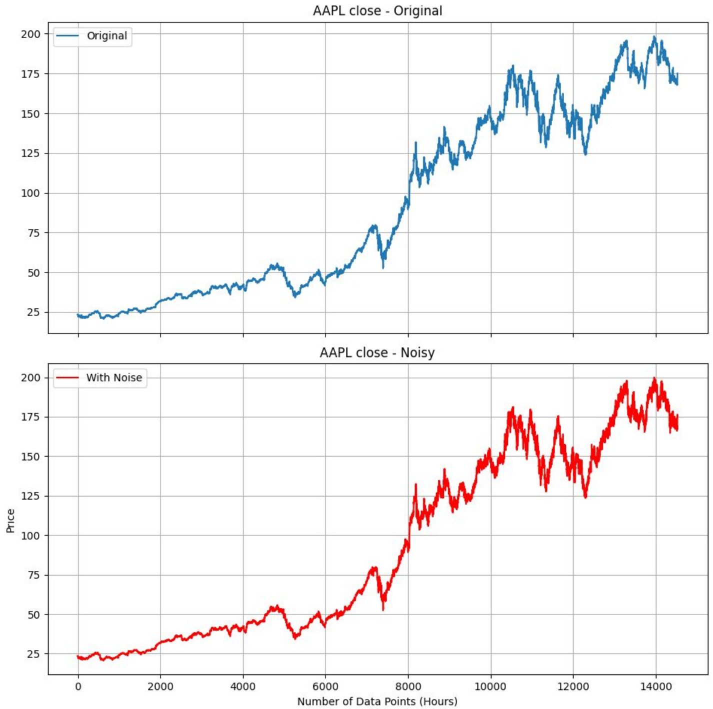


Results (Adding Noise)

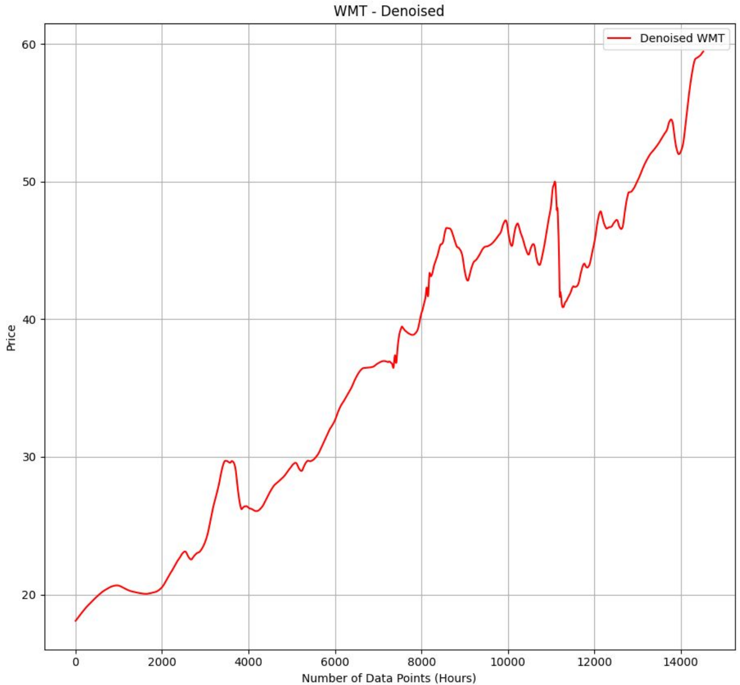




Results (Adding Noise)



Results (Example of Denoised Time Series



Results (Evaluation and Fine-Tuning)

$$MSE = \frac{1}{n}\sum_{k=1}^{n}(denoised_k - original_k)^2$$

$$SNR = 10 \cdot log_{10}\left(\frac{\sum_{k=1}^{n} original_k^2}{\sum_{k=1}^{n}(denoised_k - original_k)^2}\right)$$

Table 1. Optimized Parameters for Wavelet Denoising

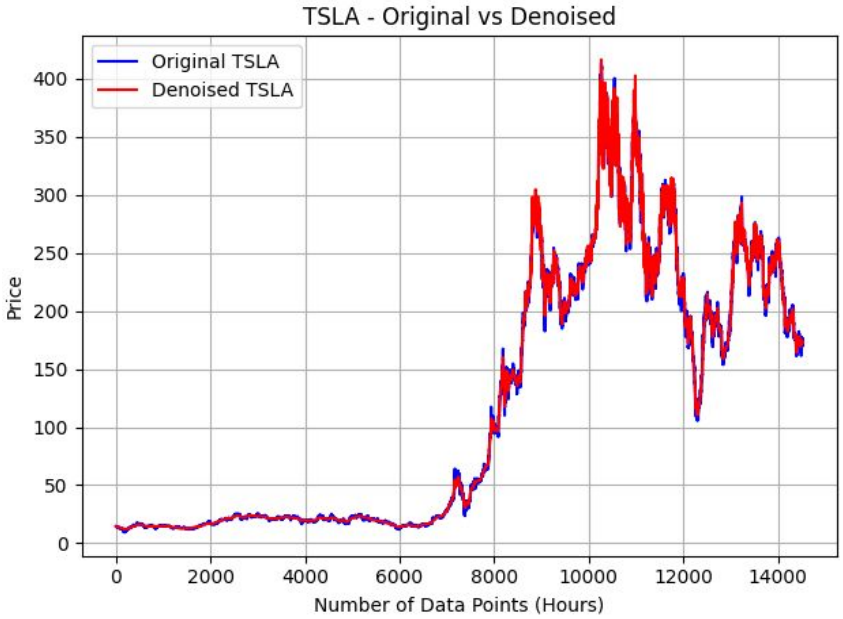
	TSLA	WMT	JNJ	INTC	AAPL
Wavelet Family and Order	Symlet 10	Symlet 11	Symlet 16	Symlet 17	Daubechies 4
Thresholding Type	Soft	Hard	Hard	Hard	Hard
Thresholding Value	8.0	2.0	8.0	2.0	8.0

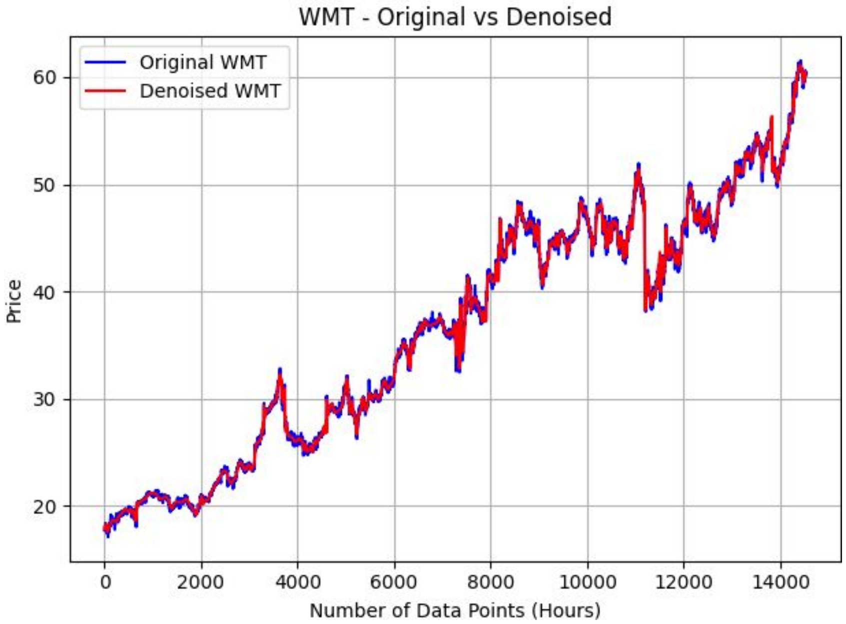
Results (Evaluation and Fine-Tuning)

Table 2. Wavelet Denoising SNR and MSE Metrics

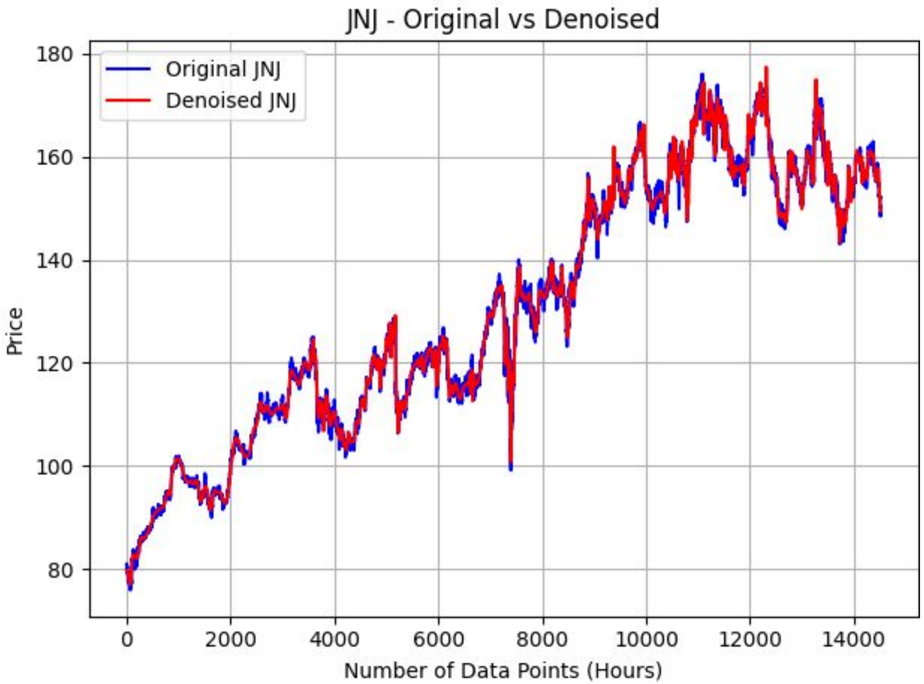
	TSLA	WMT	JNJ	INTC	AAPL
SNR	34.79 dB	42.84 dB	41.62 dB	40.85 dB	39.04 dB
MSE	8.68	0.08	1.22	0.14	1.47

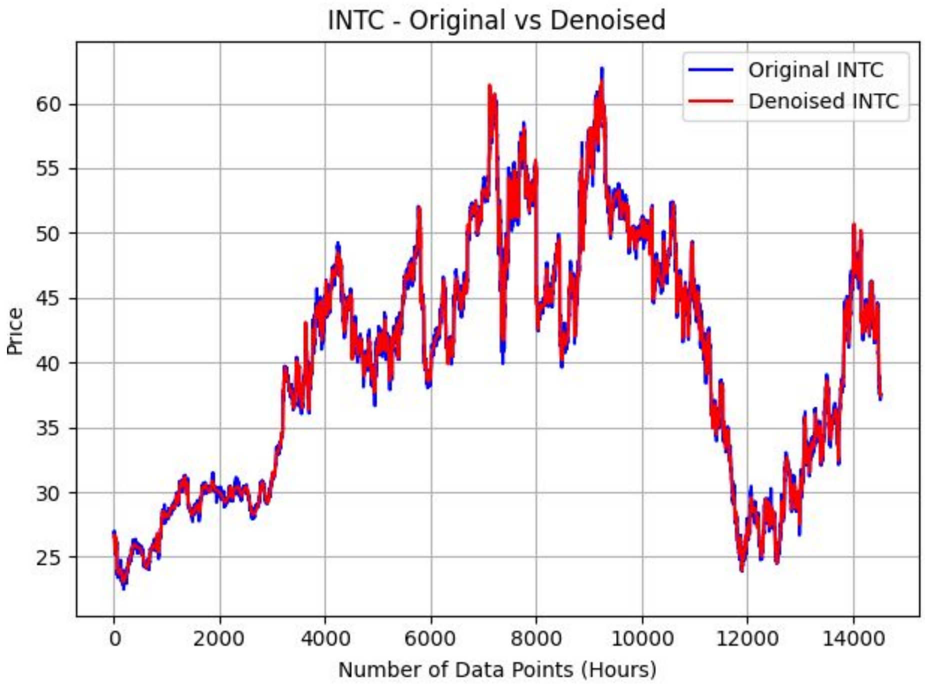
Results (Original vs Denoised)



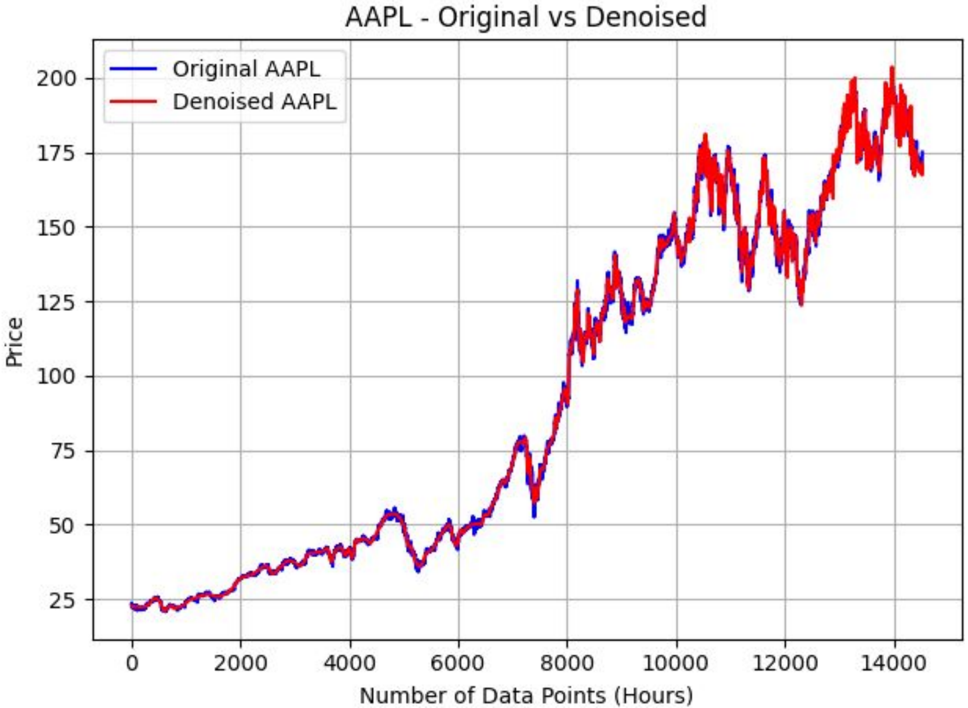


Results (Original vs Denoised)





Results (Original vs Denoised)



Results (Original vs Denoised)

Table 3. Annualized Volatility of Logarithmic Returns – Original vs Denoised

	TSLA	WMT	JNJ	INTC	AAPL
Original	52.55%	20.41%	18.15%	32.57%	26.92%
Denoised	38.73%	6.32%	8.87%	14.01%	17.77%
SNR	34.79 dB	42.84 dB	41.62 dB	40.85 dB	39.04 dB

Conclusion

- Main objective was to determine if wavelet denoising could be applied to high-frequency historical price data to identify underlying patterns or trends
- It was found that the volatility for every stock decreased after denoising was applied
  - This finding was further reinforced visually by overlaying the denoised prices on the original prices
- Additionally, a comparison was done across the 5 different stocks
  - It was found that stocks with lower volatility in the original price data had greater SNRs, and those with higher volatility tended to have worse SNRs
- For future research:
  - It would be interesting to see how the optimized parameters of the wavelets change overtime (i.e., if these parameters are stationary or non-stationary)
  - Variations of this project could be done on higher-frequency or lower-frequency data and compared to these findings
  - Traders may be interested in conducting out-of-sample studies to see how wavelet denoising may be able to predict future price movements

References

- “Chapter 3 Haar Bases.” Haar Wavelets, www.cis.upenn.edu/~cis5150/cis515-20-sl-Haar.pdf.
- “Continuous Wavelet Transform.” Stanford Center for Computer Research in Music and Acoustics, ccrma.stanford.edu/~jos/sasp/Continuous_Wavelet_Transform.html.
- “Daubechies Wavelet.” Daubechieswavelet-Wolfram Language Documentation, reference.wolfram.com/language/ref/DaubechiesWavelet.html.
- "Empirical Mode Decomposition: Theory & Applications." International Research Publication House, 2014, www.ripublication.com/irph/ijeee_spl/ijeeev7n8_14.pdf.
- Govindan, Vidya, et al. "A Hardware Trojan Attack on FPGA-Based Cryptographic Key Generation: Impact and Detection." Journal of Hardware and Systems Security, vol. 2, 2018, doi:10.1007/s41635-018-0042-5.
- “Haar Wavelet Filterbanks.” Georgia Tech School of Electrical and Computer Engineering, mdav.ece.gatech.edu/ece-6250-fall2019/notes/11-notes-6250-f19.pdf.

References

- Hall, Matt. “What Is the Gabor Uncertainty Principle?” Agile, 15 Jan. 2014, agilescientific.com/blog/2014/1/15/what-is-the-gabor-uncertainty-principle.html.
- “Introduction to Wavelet Families.” MathWorks, www.mathworks.com/help/wavelet/gs/introduction-to-the-wavelet-families.html#f3-1008627.
- Liu, Bing, and Huanhuan Cheng. “De-noising classification method for financial time series based on ICEEMDAN and wavelet threshold, and its application”. EURASIP Journal on Advances in Signal Processing, vol. 2024, no. 19, 2024, doi:10.1186/s13634-024-01115-5.
- Ke, Jinchuan, et al. "Cross-Correlation Multifractal Analysis of Technological Innovation, Financial Market and Real Economy Indices." Fractal and Fractional, vol. 7, 2023, art. 267, doi:10.3390/fractalfract7030267.
- “Python Numerical Methods.” Cubic Spline Interpolation - Python Numerical Methods, pythonnumericalmethods.studentorg.berkeley.edu/notebooks/chapter17.03-Cubic-Spline-Interpolation.html.
- Lindsay, R. W., et al. "The Discrete Wavelet Transform and the Scale Analysis of the Surface Properties of Sea Ice." IEEE Transactions on Geoscience and Remote Sensing, vol. 34, no. 3, May 1996, pp. 771-787, doi: 10.1109/36.499782.

References

- "Symlet Wavelet." Symletwavelet-Wolfram Language Documentation, reference.wolfram.com/language/ref/SymletWavelet.html.
- "The Uncertainty Principle - UGA Math." University of Georgia, www.math.uga.edu/sites/default/files/uncertainty.pdf.
- "Wavelet Signal Analyzer." MathWorks, www.mathworks.com/help/wavelet/gs/what-is-a-wavelet.html.
- "What Are Wavelet Transforms?" MathWorks, www.mathworks.com/discovery/wavelet-transforms.html#:~:text=What%20Are%20Wavelet%20Transforms%3F,transients%2C%20or%20slowly%20varying%20trends.

Python Packages Used

- Pandas
- Numpy
- Matplotlib
- PyWavelets (Pywt)