Urban Climate: Mapping Land Surface Temperature Combined with Air Temperature in Amsterdam, Netherlands

By

Pepijn Willem Herbermann

2747141

Bachelor's Thesis Extension

Earth Sciences, Economics, and Sustainability

Faculty of Science

Vrije Universiteit Amsterdam

Under the guidance of Eric Koomen and Eduardo Simao Da Graca Dias

April 2025

Table of Contents

§1: Goal

2022 was a scorching hot summer for Europe, resulting in an estimated 70.000 excess deaths due to the heat (Ballester et al., 2023). Alarmingly, a study by Heaviside et al. (2016) on the heatwaves of 2003 in the West Midlands, United Kingdom, found that half of the excess deaths in the region were due to the Urban Heat Island effect (UHI effect). Deilami et al. (2018) describe the Urban Heat Island as a phenomenon when urban areas endure higher temperatures than their non-urban surroundings. The effect is considered to have a significant influence on climate change and heat-related deaths. The UHI effect already has a major impact on the health of city inhabitants. Still, with global warming on the rise, it is becoming increasingly urgent to better understand its working mechanisms to help mitigate and prevent the effect in our urban areas. This research could help city planners develop heat-resistant neighbourhoods with, for example, more vegetation and open water, contributing to Sustainable Development Goal 11: 'Sustainable Cities and Communities'.

This thesis extension aims to gather and visualise land surface temperature (LST) and air temperature data from the region of Amsterdam, as input for the succeeding thesis. Both LST and air temperature measure temperature, but differ in what exactly they measure and how they are measured. The LST is widely monitored with thermal-infrared sensors on satellites, like Landsat-8 and MODIS, and measures the radiation from Earth's surface, such as a tree, mountain or roof. Planck's law is used to calculate the temperature by the amount of thermal-infrared radiation the surface emits (Molnár et al., 2007). The air temperature is measured with thermometers. The KNMI, the Royal Dutch Weather Institute, have thermometers scattered throughout the Netherlands, which have to satisfy certain standards like for example, a placement height of exactly 1.5 meters above the surface (*KNMI - Temperatuur*, z.d., pp. 2–3). Though there are a great deal of weather enthusiasts with automated weather stations in their backyards, whose data is broadcast to weather companies like Wunderground. A study by Meier et al. (2017) found that, despite challenges like data quality assessment, amateur weather stations provide a useful tool to asses urban climate. In the subsequent thesis, the results of this report are analysed to determine the statistical relationship between land surface temperature and air temperature. The relationship, combined with other relevant factors like NDVI and the level of urbanisation, is used to build a predictive model for air temperature based on LST.

§2: Methodology

To gather and visualise the data, the following steps are taken: first, satellite imagery from Landsat-8 and MODIS is used to generate LST maps of the area. Landsat 8 was selected for its incredible resolution (30 meter), whilst MODIS (1000 meter resolution) was chosen for its midnight overpass time, when the UHI effect tends to be strongest (Sun et al., 2020). Second, the weather data, namely the air temperature, is obtained via the Wunderground API. Third, the dates and times are matched to combine both temperatures into a single map. Amsterdam was the main focus point of this research, but including the surrounding rural areas enables a better comparison of the UHI effect between the hot city and cooler neighbouring landscape. Figure 1 on page 3 depicts our exact region of interest.

§2.1: Land surface temperature derivation

First, the Landsat-8 imagery, with just 30-meter resolution, is filtered to the summer seasons of 2022, 2023, and 2024, and masked for water and clouds. Only the summers are selected, as we are interested in the UHI effect during the highest temperature months, where the effect is greatest. Furthermore, Sobrino et al. (2008) found that water must be treated separately in the derivation of land surface temperature via satellite imagery. Therefore, a detailed water mask of Amsterdam's intricate waterways is applied. An important argument for the watermask lies in the NDVI calculations, which return strongly different values for water and land, distorting the upcoming emissivity calculations. These filters resulted in a collection of 29 images, which were all taken around 10:40 A.M. From the image collection, band 10, the thermal infrared band, is selected and plotted. All the following maps are generated by taking the median of all the individual 29 images, to keep the visualisation clean and simple. Figure 2 depicts the value stored in Band 10, transformed into the brightness temperature, which is in °K, via Planck's equation. Figure 2 might seem like the finished product, but the brightness temperature needs atmospheric corrections and adjustment for the surface emissivity (Li et al., 2013).

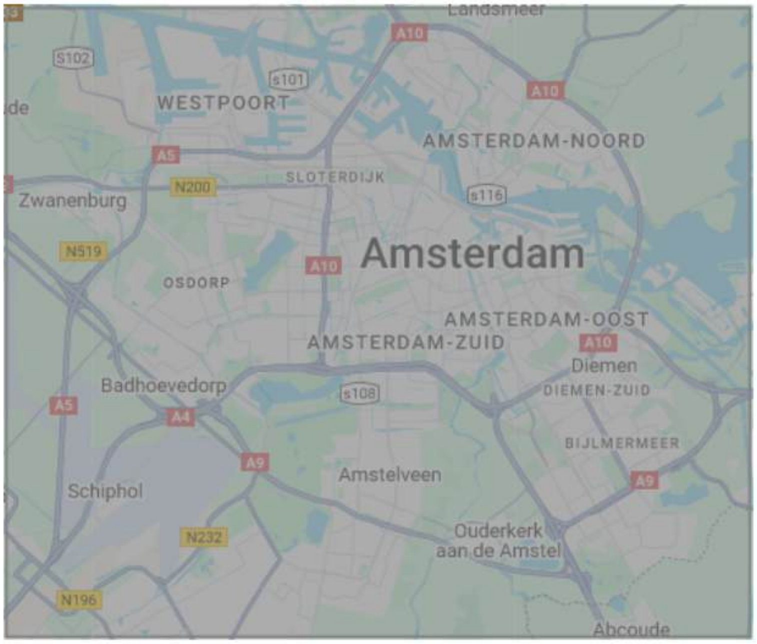


Figure 1: Region of Interest

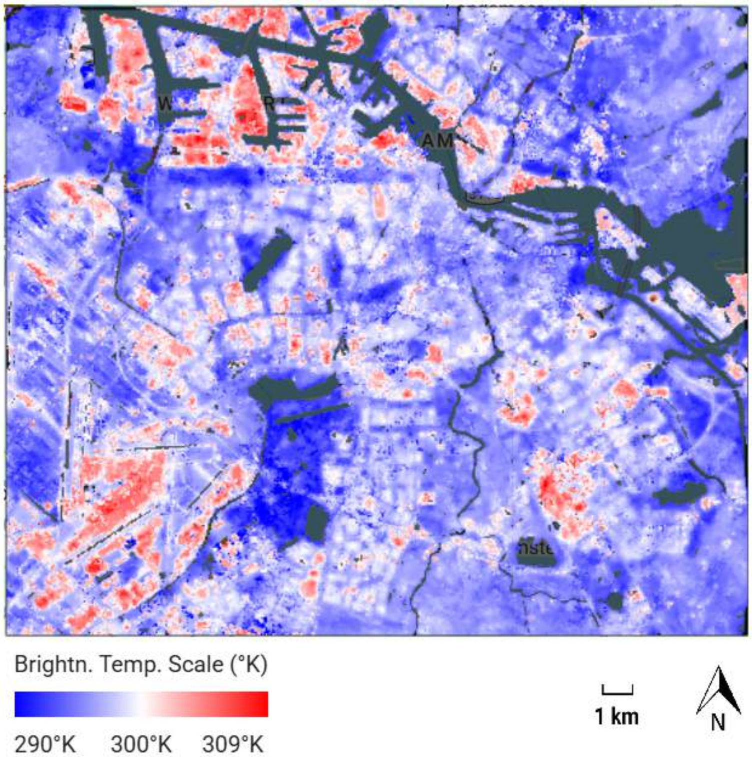


Figure 2: Landsat Brightness Temperature

The land surface emissivity (LSE), which ranges from 0 to 1, describes how well a surface emits thermal radiation in comparison to a black body. As the LST is a function of both the brightness temperature and the emissivity, you have to accurately estimate the emissivity to calculate LST. The easiest way to derive emissivity at the pixel level is as a function of the fraction of vegetation in that pixel (Malakar et al., 2018). To derive the fraction of vegetation in each pixel, the Normalized Difference Vegetation Index (NDVI) is calculated using bands 5, the near-infrared band (0.85- 0.88 micrometres), and 4, the red band (0.64- 0.67 micrometres), to fill in equation (1), of which the results are shown in Figure 1. After which, the minimum and maximum values of the NDVI are taken to calculate the fraction of vegetation in each pixel using equation (2), as shown in Figure 4. The city centre, the airport and the regular port have the lowest fraction of vegetation.

$$NDVI = \frac{(NIR - RED)}{(NIR + RED)} \tag{1}$$

$$FV = \frac{(NDVI - NDVI_{min})}{(NDVI_{max} - NDVI_{min})} \tag{2}$$

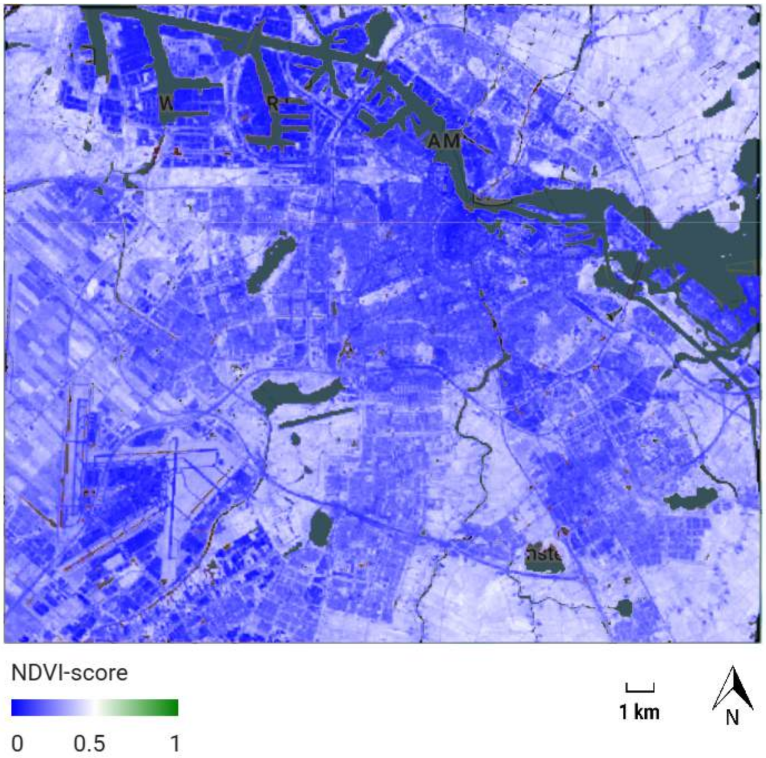


Figure 3: Normalized Difference Vegetation Index

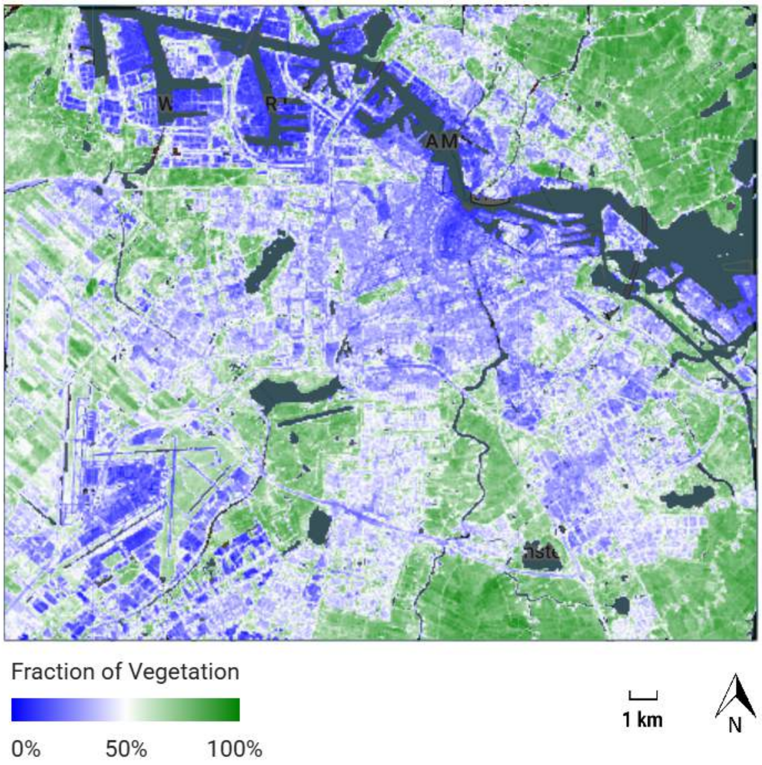


Figure 4: Fractional Vegetation

To derive emissivity from fractional vegetation, an NDVI-based LSE model is applied, equation (3). The model distinguishes three different cases: (NDVI < 0.2), which represents bare soil, or in our case, urban surface. For these NDVI values, the emissivity is calculated from the reflectance values in the red region and the two constants a_i and b_i. The next case is ($0.2 \leq$ NDVI ≤ 0.5), which represents a mix of vegetated and bare soil (or urban surface). To calculate the emissivity for this NDVI range, the fractional vegetation P_v, the bare soil (or urban surface) ε_s and vegetation emissivity ε_v are taken into account along with the cavity effect dε. Lastly, the vegetated range (NDVI > 0.5) includes the fully vegetated areas. These areas only take the vegetation emissivity ε_v and the cavity effect dε into account (Sekertekin & Bonafoni, 2020).

$$\varepsilon = \begin{cases} a_i \rho_R + b_i & \text{NDVI} < 0.2 \\ \varepsilon_v + \varepsilon_s(1 - P_v) + d\varepsilon, d\varepsilon = (1 - \varepsilon_s)(1 - P_v)F\varepsilon_V & 0.2 \leq \text{NDVI} \leq 0.5 \\ \varepsilon_V + d\varepsilon & \text{NDVI} > 0.5 \end{cases} \tag{3}$$

Yu et al. (2014) calibrated this model in their study and found the following values:

$$\varepsilon = \begin{cases} 0.984 - 0.0026\rho_R & \text{NDVI} < 0.2 \\ 0.9896P_v + 0.9747(1 - P_v) + d\varepsilon & 0.2 \leq \text{NDVI} \leq 0.5 \\ 0.9896 + d\varepsilon & \text{NDVI} > 0.5 \end{cases} \tag{4}$$

The calibrated model applied with the fractional vegetation and red-band reflectance, but for the sake of simplicity, without the cavity effect, returns the following figure:

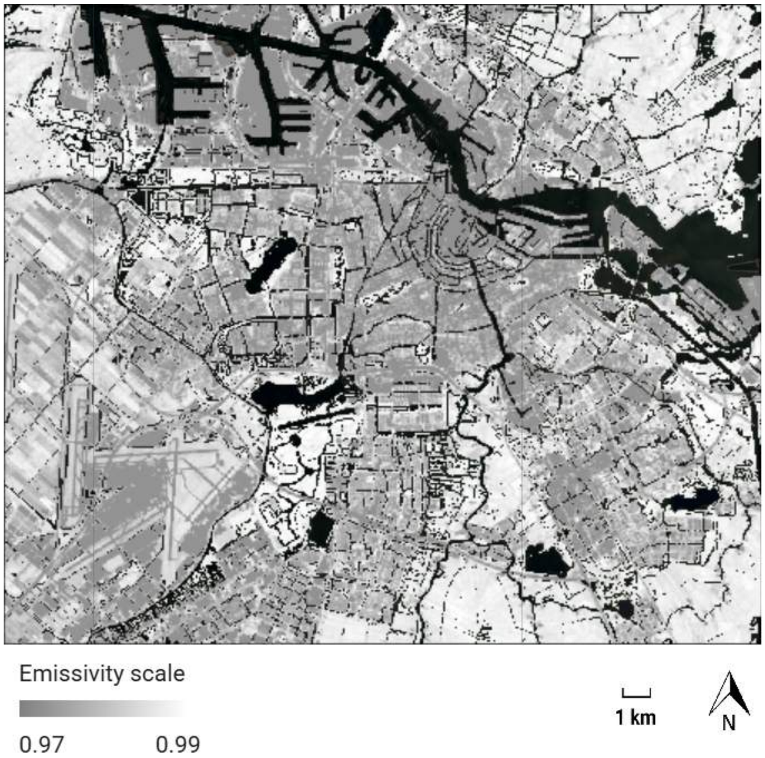


Figure 5: Land Surface Emissivity

Lastly, the LST is calculated using a simple mono-window algorithm developed by Qin et al. (2001) to estimate the LST. The mono-window algorithm is the most basic LST retrieval method because it only uses one spectral band. This method keeps the calculation simple and understandable. Equation (5), wherein lambda is the T_b brightness temperature in degrees Kelvin, λ is the wavelength for the thermal-infrared band ≈ 10.895 µm, c_2 is the second radiation constant ≈ 1.4388 cm·K, and epsilon is the land surface emissivity. Lastly, there is a correction for degrees Kelvin to degrees Celsius

$$T_S = \frac{T_b}{1 + (\frac{\lambda T_b}{c_2})\, ln(\epsilon)} - 273.15 \tag{5}$$

This resulted in a collection of 29 self-derived LST maps from the summers of 2022-2024, masked for water and clouds. All taken at the local passing time, 10:40 A.M., of Landsat-8. To better understand the relationship between LST and air temperature, an additional set of LST maps was added from the MODIS satellite. This satellite has an overpass above Amsterdam at 01:30 A.M. local time, which, as previously mentioned, offers a better insight into the UHI effect. Unlike the Landsat-8 maps, the MODIS LST maps were a finished product from NASA (*MODIS Web*, n.d.). They used a split-window algorithm to calculate LST from two near-infrared bands (band 31 and 32), to achieve higher accuracy (Wang et al., 2019). The product is already water and cloud masked. The summer seasons of the same years (2022-2024) were loaded, which resulted in a collection of 278 LST maps.

§2.2: Air temperature derivation

The Weather Underground site provides hourly summary data for thousands of amateur stations in the Netherlands. Their WunderMap was used to list all the stations in the area of interest. With the listed station ID'S and the dates of the Landsat and MODIS maps, the hourly summary data of all the matching dates were extracted with their API. The variables extracted included:
"station", "epoch", "temp_avg", "temp_high", "temp_low", "humidity_avg", "humidity_high", "humidity_low", "precip_rate", "lat", "lon", "wind_dir", "wind_speed". An overview of the variables, with units of measurement and a short explanation can be found in appendix 2.
Furthermore, the station's latitude and longitude were filtered to contain at least three decimal places, to ensure sufficient accuracy of their locations. Lastly, the hourly summary data was filtered to best match the overpass time of the Landsat-8 and MODIS satellites. 10:40 A.M. and 1:30 P.M. UTC, respectively. The results were a CSV file with summary data of 10:00-11:00 A.M for the selected variables at the 29 dates on which the Landsat-8 images were taken. And a second CSV file with summary data of 01:00-02:00 A.M. for the selected variables at the 278 dates on which the MODIS images were taken.

§2.3: Combining land surface temperature and air temperature

The LST and air temperature were combined in two ways: a visual representation and a statistical representation. The visual representation aims to keep the spatial variation by creating a map of the LST and air temperature combined. The statistical representation aims to clarify the statistical relationship by creating a scatterplot between LST and air temperature. Both visualisations were applied to Landsat-8 and MODIS in combination with their respective air temperature measurements.

The visual representation was generated by calculating the median of the air temperature measurements and the LST maps. These were overlaid to generate a map whereon the distribution of LST and air temperature were both displayed. As mentioned, the Wunderground data is not a continuous dataset, and the Landsat-8 is masked for clouds. Thus, some points on the visual representation could be the median of far fewer than 29 LST and air temperature data points.

The statistical representation required the addition of a new variable, ‘Lst_buffered,’ to the hourly summary data from Wunderground. This variable was created by carefully matching the dates of the individual LST maps to the matching dates of the air temperature measurements. Now that the date and time of both data sets were matched, a buffer of 100 meters was applied, from which the mean LST was calculated and added to the CSV file from the hourly summary data. The LST is buffered, because we believe the air temperature is dependent, not only on the LST at the exact point of measurement, but on a mean of the area’s LST. The air mixes locally and distributes the temperature of local hotspots or coolspots to its surroundings. Thus, it is necessary to look at the LST in the surroundings as well. The buffer size is an adjustable variable, and in the subsequent thesis, multiple regressions with different buffer sizes will be run to determine how local this distribution effect is. Two scatterplots, one for each satellite, were created to display the relationship between LST and air temperature.

§3: Results

Figures 6 and 7 depict the median LST maps combined with the median of the air temperature measurements from the same dates. Figure 6, derived from Landsat-8 imagery at 10:40 A.M., has a very fine resolution (30m) and shows how LST, ranging from 18 degrees to 37 degrees, can be extremely variable through space. The air temperatures also display high spatial variability, ranging from 20 to 27 degrees Celsius.

Figure 7, derived from MODIS at 1:30 A.M, has a much coarser resolution (1000m). The LST ranges from 11 to 17 degrees Celsius, and the air temperature ranges from 14 to 19 degrees Celsius, both showing less spatial variance.

Both figures show the presence of an Urban Heat Island, though the distribution varies slightly. For the Landsat-8 LST map, the harbour and the airport stand out, along with some specific neighbourhoods surrounding the city centre, whilst in the MODIS-derived LST map, the Urban Heat Island seems most present in the city centre and slowly decreases towards the rural surroundings.

In Figure 6, a correlation between the LST and the air temperature does not immediately seem evident, but in Figure 7, there does seem to be some relation between the two variables.

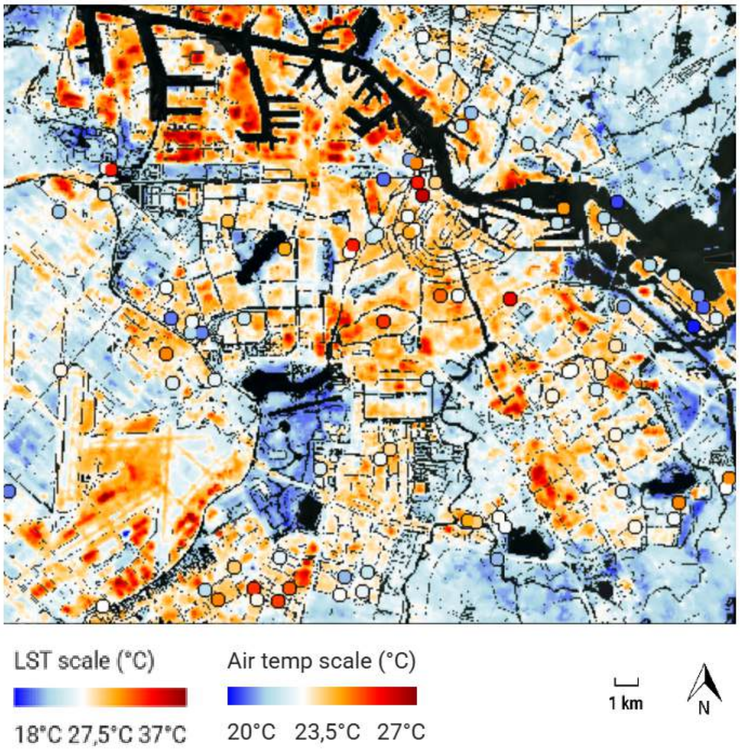


Figure 6: Landsat Land Surface Temperature and Air Temperature Measurments

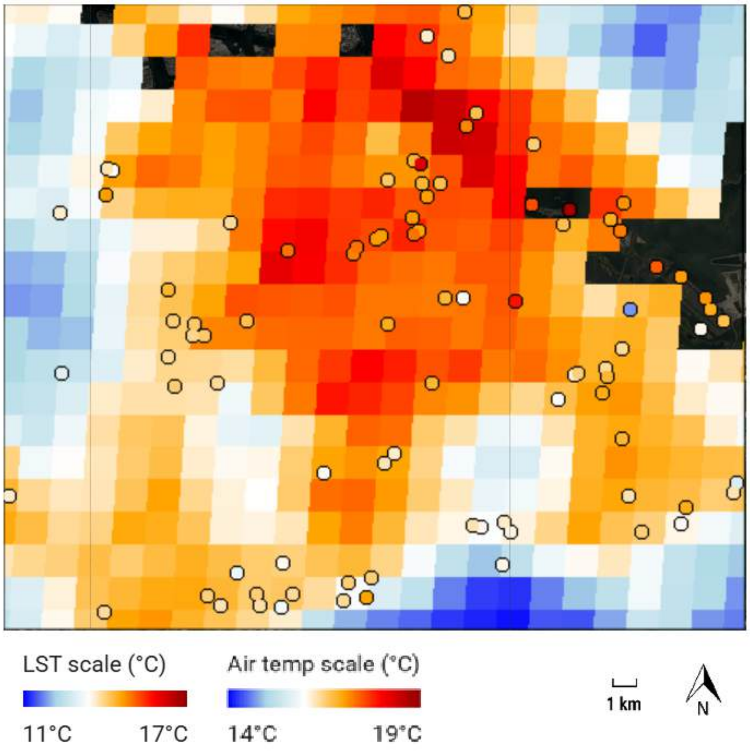


Figure 7: MODIS Land Surface Temperature and Air Temperature Measurements

The statistical relations between the LST and air temperature are depicted in Figures 8 and 9. Figure 8 depicts the relationship between average recorded air temperature from 10:00-11:00 A.M. and the local average within 100 meters—landsat-derived LST on the same date for all the stations in Amsterdam. The scatterplot shows a clear correlation between the temperatures, apart from a few heavy outliers, with very low LST.

Figure 9 is similar, but showcases the relationship between the average air temperature measured between 01:00-02:00 am and the local average within 100 meters—MODIS-derived LST. Notably, the scatter in Figure 8 is less dense and more dispersed than the scatter in Figure 9. Figure 8 is generated from 60 amateur weather stations on 29 Landsat-8 pictures, and Figure 9 is generated from 60 amateur weather stations on 278 MODIS pictures

Figure 8: Buffered LST vs. Air Temperature

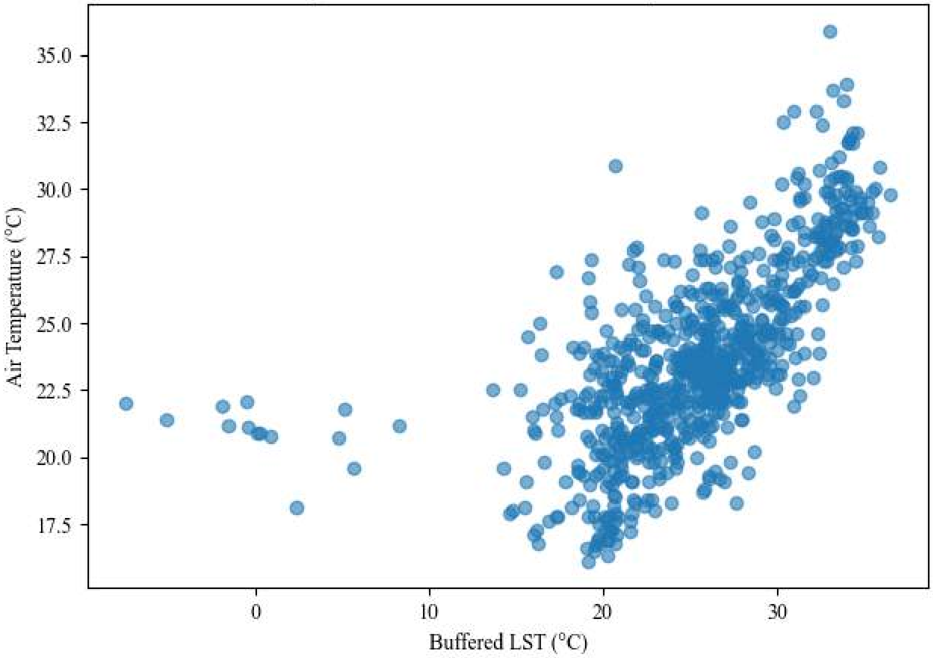
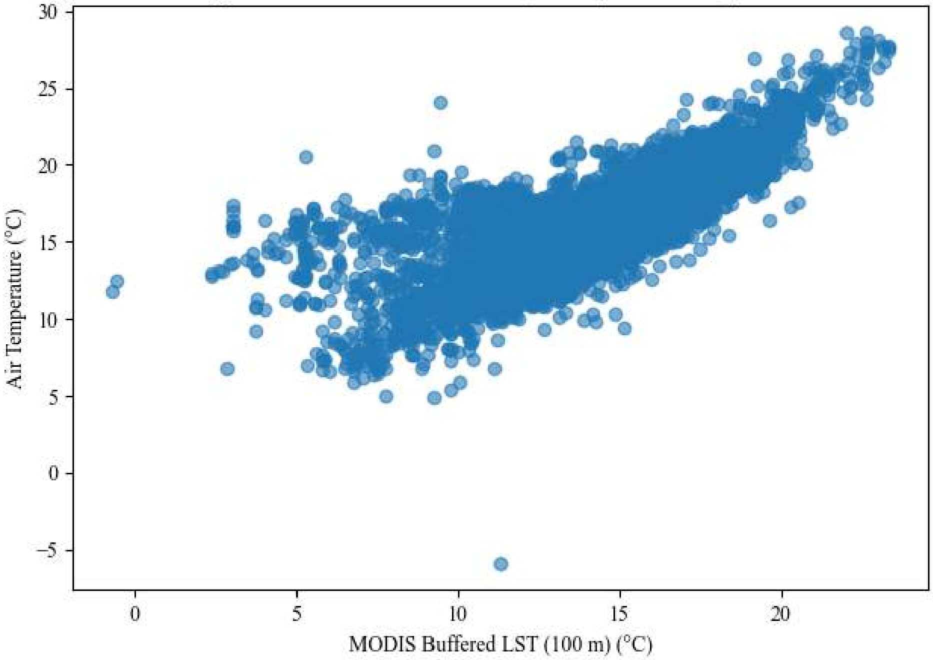


Figure 9: MODIS Buffered LST (100 m) vs. Air Temperature



§4: Discussion

In this study, Land Surface Temperature (LST) and air temperature data were gathered and visualised for the Amsterdam region, the Netherlands. Two types of LST maps were used: self-derived LST maps from Landsat-8, selected for its high resolution, and the MODIS LST product, chosen for its nighttime overpass. Both datasets were matched with air temperature measurements from nearly 90 amateur weather stations sourced from Wunderground. This resulted in visual representations showing the spatial variation of both median LST and median air temperature (see Figures 6 and 7). In the Landsat-8 map, a direct relationship between LST and air temperature was not directly evident, though some spatial correlation was visible in the MODIS figure.

Additionally, two scatterplots were created (see Figures 8 and 9), plotting air temperature against the local average LST for every single Landsat-8 and MODIS map for every single amateur station. Both plots show a strong relationship between the temperatures. The Landsat-8 scatterplot contains several very low LST values, likely due to the somewhat rough cloud filter which was applied. The plot also shows generally higher LST compared to air temperature, which is probably the result of the morning overpass time of the Landsat-8, when the land surface heats up more quickly than the air. The MODIS scatterplot includes considerably more data points, due to the greater number of LST maps available. The relationship between LST and air temperature is strong and evenly balanced. This is likely explained by the MODIS nighttime overpass at 1:30 A.M., a time when the sun no longer heats the land surface, and both air and surface temperatures cool at similar rates.

Using amateur weather stations comes with many implications. Incorrect metadata, failure of data collection and unsuitable exposure of sensors can halve the available data (Meier et al., 2017). For this study, filters, like a minimal decimal place count of three for latitude and longitude, were applied. But the failure of data collection and unsuitable exposure of sensors was unchecked. The failure of data collection in this particular study, irrelevant, as we are not performing a temporal analysis. However, the unsuitable exposure of sensors could explain part of the outliers in the scatterplots, which could lead to false relationships. Therefore, during further research, these outliers must be treated before the regression is applied.

The combination of the failure in data collection for the air temperature and the cloudmask applied to the Landsat-8 might have seriously impacted Figure 6. If too many air temperature values for certain stations were missing, and certain spots on the Landsat-8 LST map were often heavily clouded, the median of both temperatures might not be representative at all. On top, temperatures of the land surface vary strongly in space and time, the LST is known to vary by more than 10 degrees Celsius over a space of just a few centimetres (Prata et al., 1995). Therefore, it could be hard to find a spatial correlation in such a detailed map.

Furthermore, some assumptions were made in the emissivity calculation. The model distinguishes between bare soil, mixed vegetation, and vegetated surfaces. When applying this equation to the urban environment, it was assumed that bare soil has a similar emissivity to urban areas, and can therefore best be calculated by their red band radiance. Additionally, the cavity effect was simplified to a constant value. The constant was derived using the equation proposed by Sobrino and Raissouni (2000), filled in for typical urban emissivity values and fractional vegetation cover, resulting in a cavity effect correction (dε) of 0.011.

This research provides a basis for the subsequent thesis, in which the relationship between air temperature and LST will be further researched and statistically analysed. Based on the findings, a predictive model for air temperature using LST will be developed. For future research, it could be of added value to include additional explanatory variables, such as land cover and NDVI, to asses whether they can improve the understanding of the relationship between the temperatures. Another potentially relevant variable is the number of consecutive heatwave days preceding the measurements, as this may influence both surface and air temperature dynamics.

References

Ballester, J., Quijal-Zamorano, M., Turrubiates, R. F. M., Pegenaute, F., Herrmann, F. R., Robine, J. M., Basagaña, X., Tonne, C., Antó, J. M., & Achebak, H. (2023). Heat-related mortality in Europe during the summer of 2022. Nature Medicine, 29(7), 1857–1866. https://doi.org/10.1038/s41591-023-02419-z

Heaviside, C., Vardoulakis, S., & Cai, X. (2016). Attribution of mortality to the urban heat island during heatwaves in the West Midlands, UK. Environmental Health, 15(S1). https://doi.org/10.1186/s12940-016-0100-9

Deilami, K., Kamruzzaman, M., & Liu, Y. (2018). Urban heat island effect: A systematic review of spatio-temporal factors, data, methods, and mitigation measures. International Journal Of Applied Earth Observation And Geoinformation, 67, 30–42. https://doi.org/10.1016/j.jag.2017.12.009

Sobrino, J. A., Jimenez-Munoz, J. C., Soria, G., Romaguera, M., Guanter, L., Moreno, J., Plaza, A., & Martinez, P. (2008). Land Surface Emissivity Retrieval From Different VNIR and TIR Sensors. *IEEE Transactions On Geoscience And Remote Sensing*, *46*(2), 316–327. https://doi.org/10.1109/tgrs.2007.904834

Li, Z., Tang, B., Wu, H., Ren, H., Yan, G., Wan, Z., Trigo, I. F., & Sobrino, J. A. (2013). Satellite-derived land surface temperature: Current status and perspectives. *Remote Sensing Of Environment*, *131*, 14–37. https://doi.org/10.1016/j.rse.2012.12.008

Malakar, N. K., Hulley, G. C., Hook, S. J., Laraby, K., Cook, M., & Schott, J. R. (2018). An Operational Land Surface Temperature Product for Landsat Thermal Data: Methodology and Validation. *IEEE Transactions On Geoscience And Remote Sensing*, *56*(10), 5717–5735. https://doi.org/10.1109/tgrs.2018.2824828

Sekertekin, A., & Bonafoni, S. (2020). Land Surface Temperature Retrieval from Landsat 5, 7, and 8 over Rural Areas: Assessment of Different Retrieval Algorithms and Emissivity Models and Toolbox Implementation. *Remote Sensing*, *12*(2), 294. https://doi.org/10.3390/rs12020294

Yu, X., Guo, X., & Wu, Z. (2014). Land Surface Temperature Retrieval from Landsat 8 TIRS—Comparison between Radiative Transfer Equation-Based Method, Split Window Algorithm and Single Channel Method. *Remote Sensing*, *6*(10), 9829–9852. https://doi.org/10.3390/rs6109829

Qin, Z., Karnieli, A., & Berliner, P. (2001). A mono-window algorithm for retrieving land surface temperature from Landsat TM data and its application to the Israel-Egypt border region. *International Journal Of Remote Sensing*, *22*(18), 3719–3746. https://doi.org/10.1080/01431160010006971

Local Weather Forecast, News and Conditions | Weather Underground. (n.d.) Retrieved May 6, 2025. https://www.wunderground.com/

Molnár, G., Timár, G., Ferencz, C., & Lichtenberger, J. (2007). *Land surface temperature (LST) estimation algorithm for MODIS data*. https://doi.org/10.13140/2.1.3617.1207

KNMI - temperatuur. (n.d.). https://www.knmi.nl/kennis-en-datacentrum/uitleg/temperatuur#:~:text=Op%20meteorologische%20stations%20wordt%20de,hebben%20van%20een%20open%20jaloezie.

Meier, F., Fenner, D., Grassmann, T., Otto, M., & Scherer, D. (2017). Crowdsourcing air temperature from citizen weather stations for urban climate research. *Urban Climate*, *19*, 170–191. https://doi.org/10.1016/j.uclim.2017.01.006

Sun, Y., Wang, S., & Wang, Y. (2020). Estimating local-scale urban heat island intensity using nighttime light satellite imageries. *Sustainable Cities And Society*, *57*, 102125. https://doi.org/10.1016/j.scs.2020.102125

MODIS web. (n.d.). https://modis.gsfc.nasa.gov/data/dataprod/mod11.php

Wang, L., Lu, Y., & Yao, Y. (2019). Comparison of Three Algorithms for the Retrieval of Land Surface Temperature from Landsat 8 Images. *Sensors*, *19*(22), 5049. https://doi.org/10.3390/s19225049

Prata, A. J., Caselles, V., Coll, C., Sobrino, J. A., & Ottlé, C. (1995). Thermal remote sensing of land surface temperature from satellites: Current status and future prospects. *Remote Sensing Reviews*, *12*(3–4), 175–224. https://doi.org/10.1080/02757259509532285

Sobrino, J. A., & Raissouni, N. (2000). Toward remote sensing methods for land cover dynamic monitoring: Application to Morocco. *International Journal Of Remote Sensing*, *21*(2), 353–366. https://doi.org/10.1080/014311600210876

Appendix 1: GEE code for LST-mapping

```
// *** part 1: DEFINING THE AREA OF INTEREST *** //
// Create a new geometry representing the region of interest add geometry to the map
var Amsterdam = ee.Geometry.Rectangle([4.71, 52.27, 5.02, 52.43]);
Map.centerObject(Amsterdam, 11);
Map.addLayer(Amsterdam, {color: 'grey',fillColor: '',width: 2},'Region of Interest');
Map.setCenter(4.865, 52.35, 11);

// *** part 2: CREATING MASKS (WATER, SUMMERDAYS, CLOUDS) *** //
// Create a summer filter.
var sumFilter = ee.Filter.dayOfYear(172, 264);
// Generate a water mask.
var waterMask = ee.Image('projects/ee-my-pepijn/assets/water_LGN2023_wgs84');
var notWater = waterMask.neq(1);
// Function to filter out cloudy pixels.
function advancedCloudMask(image) {
  var qa = image.select('QA_PIXEL');
  var mask = qa.bitwiseAnd(1 << 3).eq(0)   // cloud
             .and(qa.bitwiseAnd(1 << 4).eq(0)) // shadow
             .and(qa.bitwiseAnd(1 << 2).eq(0)) // cirrus
             .and(qa.bitwiseAnd(1 << 5).eq(0)); // snow
  return image.updateMask(mask);
}
// *** part 3: CREATE LST MAP WITH LANDSAT8 DATA *** //
// Load Landsat collection and apply filters
var col = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA')
    .filterBounds(Amsterdam)
    .filterDate('2022-01-01', '2025-01-01')
    .filter(sumFilter)
    .map(advancedCloudMask);
// Calculate median image
var medianImage = col.median().clip(Amsterdam).updateMask(notWater);
// Compute NDVI
var ndvi = medianImage.normalizedDifference(['B5', 'B4']).rename('NDVI')
    .updateMask(notWater);
Map.addLayer(ndvi, {
  min: 0,
  max: 1,
  palette: ['blue', 'white', 'green']
}, 'NDVI');
// Compute FV
var ndviMin = 0.2;
var ndviMax = ee.Number(ndvi.reduceRegion({
  reducer: ee.Reducer.max(),
  geometry: Amsterdam,
  scale: 30,
  maxPixels: 1e9
}).get('NDVI'));
var fv = ndvi.subtract(ndviMin).divide(ndviMax.subtract(ndviMin)).rename('Pv')
    .clamp(0, 1);
Map.addLayer(fv, {
  min: 0,
  max: 1,
  palette: ['blue', 'white', 'green']
}, 'Fractional Vegetation (FV)');
// Compute emissivity
var red = medianImage.select('B4').multiply(0.0001).rename('RED');
var emissivityRed = red.multiply(-0.047).add(0.973).rename('EM10_Red');
var epsV = 0.9863;
var epsS = 0.9668;
var C = 0.011;
var emissivityPv = ee.Image.constant(epsV).multiply(fv)
  .add(ee.Image.constant(epsS).multiply(ee.Image.constant(1).subtract(fv)))
  .add(C).rename('EM10_Pv');
var em10 = emissivityPv.where(ndvi.lt(ndviMin), emissivityRed).rename('EM10');
Map.addLayer(em10, {
  min: 0.97,
  max: 0.99,
  palette: ['grey', 'white']
}, 'Emissivity (EM10)');
// Reload collection for individual LST computation with Date Property
var colLST = ee.ImageCollection('LANDSAT/LC08/C02/T1_TOA')
    .filterBounds(Amsterdam)
    .filterDate('2022-01-01', '2025-01-01')
    .filter(sumFilter)
    .map(advancedCloudMask)
    .map(function(image) {
      var date = ee.Date(image.get('system:time_start')).format('YYYY-MM-dd');
      return image.set('date', date);
    });
// Select the latest image per day
var distinctDates = colLST.distinct(['date']);
var latestPerDay = ee.ImageCollection(distinctDates.map(function(dateFeature) {
  var date = ee.String(dateFeature.get('date'));
  var dailyImages = colLST.filter(ee.Filter.eq('date', date));
  return dailyImages.sort('system:time_start', false).first();
}));
// Function to Compute LST per Image Using Shared Emissivity
var computeLST = function(image) {
  var thermal = image.select('B10')
    .clip(Amsterdam)
    .updateMask(notWater);
  var lst = thermal.expression(
    '(Tb / (1 + (0.001145 * (Tb / 1.438)) * log(Ep))) - 273.15', {
      'Tb': thermal.select('B10'),
      'Ep': em10.select('EM10')
    }).rename('LST').updateMask(notWater);
  return image.addBands(lst);};
// Apply LST calculation to filtered collection
var lstCollection = latestPerDay.map(computeLST);
print('LST Individual Images (One per Day):', lstCollection);
// Median LST from the processed collection
var lstMedian = lstCollection.select('LST').median();
Map.addLayer(lstMedian, {
  min: 18,
  max: 37,
  palette: ['blue', 'lightblue', 'white', 'orange', 'red', 'darkred']
}, 'LST Landsat (Median)');
// *** PART 4: Load night time LST map from modis aqua 1km res. *** //
var modisLST = ee.ImageCollection("MODIS/061/MYD11A1")
  .filterBounds(Amsterdam)
  .filter(sumFilter)
  .filterDate('2022-01-01', '2025-01-01')
  .select('LST_Night_1km')
  .map(function(img) {
    return img
      .multiply(0.02)
      .subtract(273.15)
      .copyProperties(img, ["system:time_start"]);
  });
// Print modisLST to console
print(modisLST, 'modisLST')
// Take median composite for testing
var medianLST = modisLST.median().clip(Amsterdam);
// Visualization parameters
var visParams = {  min: 11,max: 17,  palette: ['blue', 'lightblue', 'white', 'orange', 'red', 'darkred']};
// Add to map
Map.addLayer(medianLST, visParams, 'MODIS Aqua LST (Night, °C)');
```

```
// *** part 5A: LOADING/VISUALIZING AIR TEMP POINT AND LST (similair for MODIS) *** //
// Load your air temperature measurements
var airTempPointsLandsat =
ee.FeatureCollection('projects/ee-my-pepijn/assets/cleaned_station_data');
// Print to see properties and structure
print('Air Temp Points Landsat:', airTempPointsLandsat);
// Add a dateOnly property for the airTempPoints featurecollection for the matching process
airTempPointsLandsat = airTempPointsLandsat.map(function(f) {
  var dtString = ee.String(f.get('datetime'));
  var dateOnly = ee.Date.parse('YYYY-MM-dd HH:mm:ss', dtString, 'UTC')
    .format('YYYY-MM-dd');
  return f.set('date_only', dateOnly);
// Group by station ID and calculate median temp_high
var medianPerStation = airTempPointsLandsat.reduceColumns({
  selectors: ['station', 'temp_high'],
  reducer: ee.Reducer.median().group({
    groupField: 0,
    groupName: 'station_id'})
});
// Get the results as a list of dictionaries
var medianList = ee.List(medianPerStation.get('groups'));
// Convert to a FeatureCollection with geometry
var stationFC = ee.FeatureCollection(medianList.map(function(dict) {
  dict = ee.Dictionary(dict);
  var id = dict.get('station_id');
  var medianVal = dict.get('median');
  // Find a feature to get the geometry
  var feature = airTempPointsLandsat.filter(ee.Filter.eq('station', id)).first();
  var geom = feature.geometry();
  return ee.Feature(geom, {
    'station_id': id,
    'median_temp_high': medianVal});
// print it for viewing
print('Median high_temp points Landsat',stationFC)
// Add to map with style
var stationImage = airTempPointsLandsat.reduceToImage({
  properties: ['median_temp_high'],
  reducer: ee.Reducer.first()
});
// Style similar to LST
var visParams = {
  min: 20,
  max: 27,
  palette: ['blue', 'lightblue', 'white', 'orange', 'red', 'darkred']  // Color gradient};
```

```
// Rasterize the air temperature points (median_high_temp)
var tempImage = stationFC.reduceToImage({
  properties: ['median_temp_high'],  // Property to be rasterized
  reducer: ee.Reducer.first()  // Just take the first value (since we already computed median per station)
});
// Apply a 300-meter buffer for visibility
var bufferedTempImage = tempImage.focal_max(300, 'circle', 'meters');
// Add black outline by creating a slightly larger buffer
var outlineBuffer = tempImage.focal_max(380, 'circle', 'meters');
// Add the black outline to the map (slightly larger buffer to act as outline)
Map.addLayer(outlineBuffer, {
   palette: ['black']
}, 'Landsat Black outline buffer');
// Style the rasterized temperature image
Map.addLayer(bufferedTempImage, visParams, 'Landsat Buffered Median temp_high');

// *** PART 9A: Matching air temp points to buffered Landsat LST (similair for MODIS) *** //
// Create an Adjustable buffer size in meters
var bufferSize = 100;
// Load air temperature measurement points and add a 'date_only' property
var airTempPoints = ee.FeatureCollection('projects/ee-my-pepijn/assets/cleaned_station_data')
  .map(function(f) {
    var date = ee.Date
      .parse('YYYY-MM-dd HH:mm:ss', ee.String(f.get('datetime')), 'UTC')
      .format('YYYY-MM-dd');
    return f.set('date_only', date);
  });
// Define date range of interest
var startDate = '2022-01-01';
var endDate   = '2024-12-31';
// Filter the LST collection to that date range and select only the LST band
var lstImages = lstCollection
  .filter(ee.Filter.and(
    ee.Filter.gte('date', startDate),
    ee.Filter.lte('date', endDate)))
  .select('LST');
// For each image/date, buffer the matching station points and compute mean LST
var bufferedByDate = lstImages.map(function(image) {
  // Take the date string from the image metadata
  var date = image.get('date');
  // Select only those station points on that same date, and buffer them
  var pts = airTempPoints
    .filter(ee.Filter.eq('date_only', date))
    .map(function(f) {
      return ee.Feature(
        f.geometry().buffer(bufferSize),
        f.toDictionary() ); });
  // Compute mean LST within each buffer, name the output 'buffered_LST'
  return image
    .reduceRegions({
      collection: pts,
      reducer: ee.Reducer.mean().setOutputs(['buffered_LST']),
      scale: 30})
    .map(function(f) {
      // Preserve the date on each feature
      return f.set('date_only', date);});
});
// Flatten the per-image collections into one FeatureCollection
var allBuffered = ee.FeatureCollection(bufferedByDate).flatten();
// Create table with only station, date, and buffered_LST
var exportTable = allBuffered.map(function(f) {
  return ee.Feature(null, f.toDictionary([
    'station',
    'date_only',
    'buffered_LST'
  ]));
});
// Export the complete table in one CSV
Export.table.toDrive({
  collection: exportTable,
  description: 'Buffered_LST_2022_2024',
  fileFormat: 'CSV',
  fileNamePrefix: 'Buffered_LST_2022_2024'});
```

Appendix 2: Table of Variables

Field Name	Description
epoch	Time in UNIX seconds
humidityAvg	Average Humidity of the period
humidityHigh	Highest Humidity of the period
humidityLow	Lowest Humidity of the period
lat	Latitude of PWS
lon	Longitude of PWS
obsTimeLocal	Time observation is valid in local apparent time by timezone
stationID	ID as registered by wunderground.com
winddirAvg	Wind direction average of the period
precipRate	Rate of precipitation - instantaneous precipitation rate. How much rain would fall if the precipitation intensity did not change for one hour
precipTotal	Accumulated Rain for the day ffrom midnight to present in defined unit of measure
qcStatus	Quality control indicator: -1: No quality control check performed 0: This observation was marked as possibly incorrect by our quality control algorithm 1: This observation passed quality control checks
tempAvg	Temperature average of the period
tempHigh	High Temperature of the period
tempLow	Low Temperature of the period
windspeedAvg	Wind speed average of the period

From: (Local Weather Forecast, News And Conditions | Weather Underground, n.d.)