University of Engineering and Management

Mathematics – III (BSM301)

By

Subrata Saha

Module 1: Timeline – 15.08.2026

Why this matter

IoT: sensor packet loss and node failure counts are classic Binomial/Poisson events; sensor battery lifetimes and inter-arrival times of readings follow Exponential distributions.

Cyber Security: number of intrusion attempts in a fixed window (Poisson), pass/fail outcomes of authentication attempts (Binomial), and time between successive attacks (Exponential) all drive alerting thresholds.

Blockchain: block arrivals in Proof-of-Work chains are commonly modelled as a Poisson process; transaction confirmation waiting time is Exponential under that model.

Q1. Binomial and Poisson fundamentals

(a) A wireless sensor node transmits a reading every minute; each transmission is corrupted independently with probability 0.02. Find the probability that, out of 50 transmissions, at most 2 are corrupted, using the Binomial distribution.

(b) Verify your answer to (a) using the Poisson approximation to the Binomial. State the condition under which this approximation is valid and comment on how close the two answers are.

Q2. Modelling failed login attempts as a Binomial process

A login server logs authentication attempt. Historically, 3% of attempts are malicious (failed credential-stuffing tries). In a batch of 200 attempts, let X be the number of malicious attempts.

(a) Write down the exact distribution of X and compute $P(X \geq 10)$ analytically.

(b) Write an Python script that simulates 100,000 batches of 200 attempts (Bernoulli trials with $p = 0.03$), estimates $P(X \geq 10)$ empirically, and compares it to your analytical answer.

(c) Using the mean and variance formulae for the Binomial distribution, propose a simple threshold rule (e.g., mean + k·SD) an intrusion detection system could use to flag an unusually malicious batch, and justify your choice of k.

Q3. Poisson model for block arrivals in a blockchain network

A Proof-of-Work blockchain is tuned so that blocks arrive at an average rate of 1 block every 10 minutes ($\lambda = 6$ blocks/hour).

(a) Find the probability that exactly 8 blocks are mined in a given hour, and the probability that more than 10 blocks are mined in that hour.

(b) Simulate one week of block arrivals (in R or Python) using the Poisson distribution per hour, and plot the observed hourly block counts as a histogram against the theoretical Poisson pmf.

Q4. Exponential distribution for sensor node lifetime

The battery lifetime of a low-power IoT temperature sensor is modelled as Exponential with a mean lifetime of 500 days.

(a) Find the probability that a randomly deployed sensor survives beyond 600 days, and the probability it fails within the first 100 days.

(b) Using the memoryless property, find the probability that a sensor which has already survived 400 days survives at least 200 more days. Verify this equals your answer for surviving beyond 200 days from deployment.

(c) For a network of 100 identical sensors deployed simultaneously, write a script to simulate their failure times and estimate, empirically, the expected number of sensors still functioning after one year (365 days). Compare with the theoretical expectation.

Module 2: Timeline – 27.08.2026

Why this matters

IoT: raw sensor output needs calibration against a known reference — least-squares straight-line and parabola fits correct for drift and non-linearity.

Cyber Security: growth of malware variants, phishing volume, or vulnerability disclosures over time is often modelled with exponential curve fitting to project short-term trends.

Blockchain: network hash rate, active-address count, or gas-fee trends are frequently fitted with straight-line or exponential curves to spot regime changes.

Q1. Straight-line and parabola fitting (hand computation)

(a) Given five (x, y) data points of your choosing (or use x = 1..5 with y = 2, 5, 10, 17, 26), fit a straight line $y = a + bx$ by the method of least squares, showing the normal equations explicitly.

(b) Using the same data, fit a second-degree parabola $y = a + bx + cx^2$ and compare the sum of squared residuals with the straight-line fit. Which model fits better, and why might that matter when the underlying process is not actually linear?

Q2. Calibrating a low-cost IoT temperature sensor

A low-cost sensor's raw ADC output was recorded alongside readings from a calibrated reference thermometer at 10 temperatures spanning 0–100 °C (simulate this data: let reference temperature T be 0,10,...,90 and raw output $R = 2.3 \cdot T + 15 + \text{noise}$, noise ~ Normal(0, 3)).

(a) Fit a least-squares straight line $R = a + bT$ to your simulated data by hand or by matrix computation, and report a, b.

(b) Use the fitted line to build a calibration function that converts a new raw reading back into an estimated temperature. Test it on 3 held-out simulated points and report the calibration error.

(c) Discuss what evidence in your residual plot would tell you a straight-line calibration is inadequate and a quadratic fit is needed instead.

Q3. Fitting exponential growth to a vulnerability-disclosure trend

Simulate 12 months of cumulative disclosed vulnerabilities in a software ecosystem, where the true underlying trend is exponential: $N(t) = N0 \cdot e^{kt}$ with $N0 = 50$, $k = 0.15$, plus small multiplicative noise.

(a) Linearize the exponential model by taking logs, and fit a straight line by least squares to ln(N) vs t to recover estimates of N0 and k.

(b) Use the fitted model to forecast the cumulative disclosure count 3 months beyond your data, and give a one-sentence caveat about the risk of extrapolating an exponential trend too far forward.

Module 3: Time line - 15.09.2026

Why this matters

IoT: it is often infeasible to poll every sensor in a large deployment, so estimates of network-wide quantities (mean temperature, mean latency) rely on sampling theory and standard error.

Cyber Security: security teams sample network traffic or log entries rather than inspecting every packet; sampling distributions of proportions underlie estimates of the malicious-traffic rate.

Blockchain: auditors and analytics tools sample transactions from a block or mempool to estimate network-wide statistics such as average fee or the randomness quality of generated nonces.

Q1. Sampling distribution of the mean and standard error

(a) A population has mean $\mu = 40$ and standard deviation $\sigma = 8$. For random samples of size $n = 64$ drawn without replacement from a large population, find the standard error of the sample mean and the probability that a sample mean exceeds 42.

(b) State the Central Limit Theorem and explain, without a full proof, why it justifies treating the sampling distribution of the mean as approximately Normal even when the population itself is not Normal.

Q2. Estimating mean network latency from a sensor swarm

A fleet of 5,000 IoT sensors reports network latency, with true (unknown to your analysis) population parameters you should simulate as Normal(mean = 120 ms, sd = 25 ms).

(a) Draw a single random sample of size 50 from the simulated population and compute the sample mean and its standard error.

(b) Repeat the sampling process 1,000 times, plot the resulting distribution of sample means, and compare its spread to the theoretical standard error from part (a).

(c) Using your simulation, comment on how the spread of the sampling distribution would change if the fleet used sample size 200 instead of 50, and why this matters when deciding how many sensors a monitoring dashboard needs to poll each cycle.

Q3. Sampling proportion of malicious packets in network traffic

Suppose the true proportion of malicious packets in a network's traffic is p = 0.04 (unknown to the analyst in practice). A security tool samples 500 packets per minute.

(a) Find the standard error of the sample proportion and the probability that a one-minute sample shows a malicious-packet proportion above 0.06 (a likely false-alarm-triggering level).

(b) Simulate 10,000 one-minute samples in R or Python, plot the empirical sampling distribution of the sample proportion, and compare the empirical probability of exceeding 0.06 to your analytical answer in (a).

(c) Discuss the practical trade-off a security team faces between sample size (packets inspected per minute) and how quickly a genuine spike in malicious traffic can be detected with statistical confidence.

Q4. Chi-square test of randomness in blockchain nonce generation

A blockchain's Proof-of-Work mining process searches for a nonce value; the last hexadecimal digit of successful nonces should, if the search process is unbiased, be uniformly distributed over the 16 possible digits.

(a) Simulate 1,600 mined-block nonces under the null hypothesis of uniform digit distribution, tabulate the observed frequency of each last-digit value, and carry out a chi-square goodness-of-fit test against the uniform distribution.

(b) Now simulate a biased nonce generator (e.g., digit 0 occurs with double the probability of the others) and repeat the chi-square test. Report the test statistic, degrees of freedom, and conclusion in both cases.

(c) Explain briefly why a mining pool or auditor might care about this kind of randomness test in a real blockchain system.

Module 4: Time line - 28.09.2026

Why this matters for IoT

IoT: maximum likelihood estimation is used to fit failure-rate and lifetime models to observed sensor field data, and confidence intervals bound the uncertainty in fleet-wide reliability estimates.

Cyber Security: confidence intervals quantify uncertainty in estimated breach rates or detection rates from limited incident data, which matters for risk reporting.

Blockchain: point and interval estimation of block time or transaction-fee parameters informs network-performance dashboards and fee-market analysis.

Q1. Point and interval estimation

(a) A sample of 36 observations has sample mean 55 and sample standard deviation 9. Construct a 95% confidence interval for the population mean.

(b) Explain the difference between a biased and an unbiased estimator, and show that the sample variance formula with divisor $(n - 1)$, rather than n, is required for unbiasedness (state the key step of the derivation; a full proof is not required).

Q2. Maximum likelihood estimation of IoT sensor failure rate

Field data from 40 deployed sensors gives their observed lifetimes (in days) before failure; simulate this data by drawing 40 samples from an Exponential distribution with true rate $\lambda = 1/400$.

(a) Derive the maximum likelihood estimator of λ for an Exponential distribution in general (show the derivation), then apply it to your simulated sample to get an MLE estimate $\hat{\lambda}$.

(b) Construct an approximate 95% confidence interval for the mean lifetime $1/\lambda$ using your sample, and check whether the true mean lifetime (400 days) falls inside your interval.

(c) If only 10 sensors had been observed instead of 40, how would you expect the width of the confidence interval to change? Verify by re-running your simulation with $n = 10$.

Q3. Confidence interval for a data-breach detection rate

Out of 150 simulated intrusion attempts fed to a detection system, simulate the number correctly flagged using a true detection probability of $p = 0.88$.

(a) Compute the point estimate of the detection rate $\hat{p}$ and construct a 95% confidence interval for the true detection probability p.

(b) Suppose management wants the margin of error of this confidence interval to be no more than 0.03. Using the standard sample-size formula for a proportion, determine the minimum number of test intrusions that should be run, and explain briefly why a conservative choice of p (e.g., $p = 0.5$) is sometimes used when planning such a test in advance.

Q4. Estimating mean block confirmation time in a blockchain network

Simulate 60 observed block confirmation times (in seconds) for a blockchain network, drawn from a Normal distribution with an unknown-to-you true mean and standard deviation of your choosing (document your chosen 'true' values so a grader can check your simulation).

(a) Compute the sample mean and sample standard deviation, then construct a 99% confidence interval for the true mean confirmation time.

(b) Using Maximum Likelihood Estimation for a Normal distribution, show that the MLE of the mean coincides with the sample mean, and state (without full derivation) why the MLE of the variance is biased while the usual sample variance (divide by $n - 1$) is not.

(c) Discuss, in one paragraph, why a network-monitoring dashboard reporting only a point estimate of block time (with no interval) could mislead users about network reliability.

Module 5: Time line - 10.10.2026

Why this matters

IoT: hypothesis testing checks whether a firmware update or hardware revision actually changed sensor accuracy, power draw, or latency, rather than the observed difference being due to chance.

Cyber Security: hypothesis tests compare detection rates of two IDS configurations or test whether an observed spike in traffic is statistically significant, guiding whether an alert warrants escalation.

Blockchain: hypothesis and chi-square tests are used to compare transaction throughput or confirmation times across competing network configurations or forks, and to test independence between transaction attributes.

Q1. Large-sample and small-sample tests (hand computation)

(a) Two independent samples of size 100 and 120 have sample means 48 and 45 with known population standard deviations 6 and 7 respectively. Test at the 5% level whether the two population means differ.

(b) A small sample of 12 observations has sample mean 30 and sample standard deviation 4. Test, at the 5% level, the hypothesis that the population mean is 32, using the appropriate small-sample (t) test, and state your conclusion.

Q2. Testing whether a firmware update changed sensor accuracy

Simulate 30 accuracy-error readings (in %) from sensors running old firmware, Normal(mean = 2.5, sd = 0.6), and 30 readings from sensors running new firmware, Normal(mean = 2.1, sd = 0.6).

(a) State appropriate null and alternative hypotheses for testing whether the firmware update reduced the mean accuracy error, and carry out a two-sample test for difference of means at the 5% significance level.

(b) Report the test statistic, the critical value (or p-value), and your conclusion in plain language a firmware engineer (not a statistician) would understand.

(c) If the sample sizes had been 8 and 8 instead of 30 and 30, which test would be more appropriate, and why?

Q3. Comparing two intrusion detection systems and testing traffic independence

System A correctly detects 178 out of 200 simulated attacks; System B correctly detects 165 out of 180 simulated attacks (simulate both from Binomial with the given n and observed count).

(a) Test, at the 5% level, whether the two systems' true detection rates differ, using a test for difference of proportions.

(b) A separate dataset cross-tabulates traffic source (internal/external) against outcome (benign/malicious) for 400 packets. Construct a simulated 2×2 contingency table with plausible counts and use a chi-square test of independence to test whether traffic source and maliciousness are associated.

(c) Explain what it would mean, operationally, if the chi-square test in (b) rejected independence — how might a security analyst use that finding?

Q4. Testing for a difference in block confirmation times across two network configurations

Simulate confirmation times (seconds) for 15 transactions on Network Configuration 1, Normal(mean = 12, sd = 2.5), and 15 transactions on Network Configuration 2, Normal(mean = 10, sd = 2.5).

(a) Using the appropriate small-sample test for difference of means, test at the 5% level whether Configuration 2 gives significantly faster confirmation, stating your null and alternative hypotheses clearly.

(b) Separately, simulate the last digit of 500 transaction hash values (0–9) under two scenarios — a fair hashing scheme and a biased one — and use a chi-square goodness-of-fit test to determine which scenario your data came from without being told in advance (i.e., run the test blind and report your conclusion).

(c) Summarize, in a short paragraph, why both hypothesis testing and chi-square goodness-of-fit tests are useful tools for auditing the statistical health of a blockchain network's operational data.